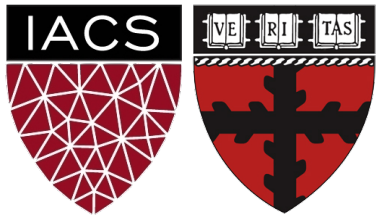
Lecture 23: Language RepresentAti.ns

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 specific task

Language Modelling

$$\theta\left("I\ love\ CS109B" | \alpha, \beta\right)$$

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

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

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

$|V|$ = 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

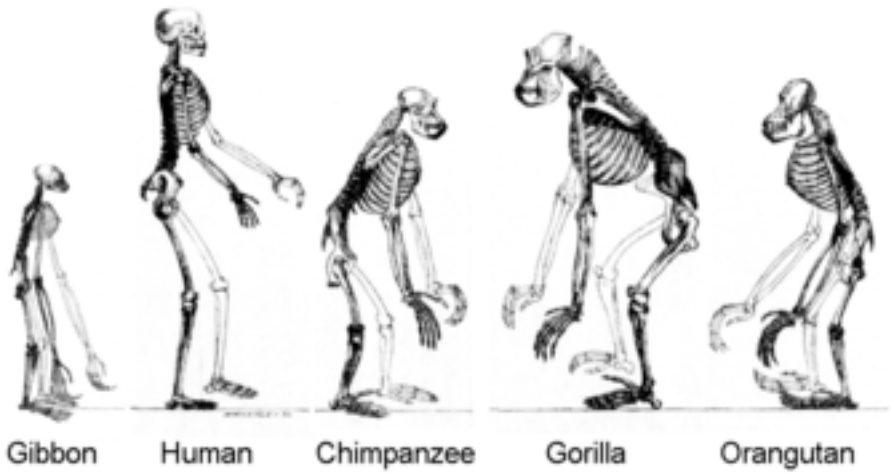
- Recap where we are
- Representing Language
  - What
  - How
  - Modern Breakthroughs

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)

Slide adapted from or inspired by Chris Manning and Richard Socher

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

Lexemes

Morphology

phonology

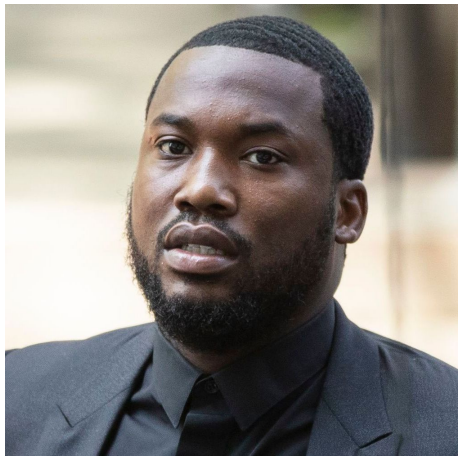
phonetics

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

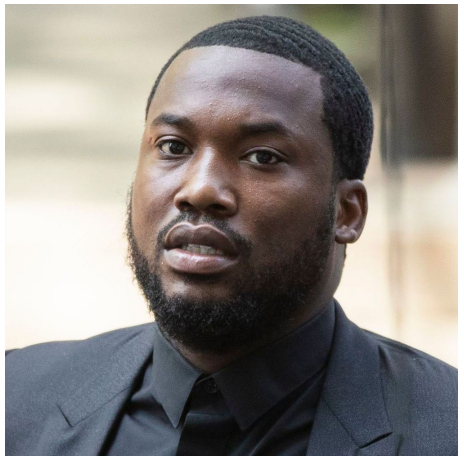
speech

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

phonetics

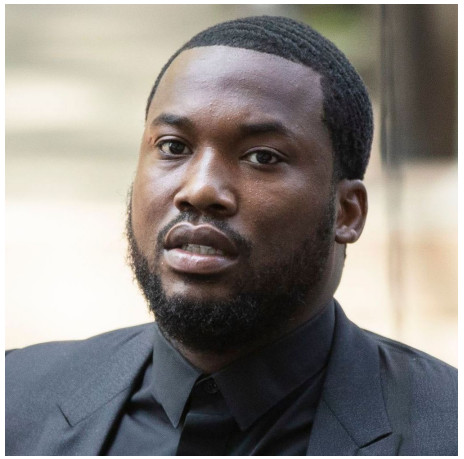
speech

orthography

text

- 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

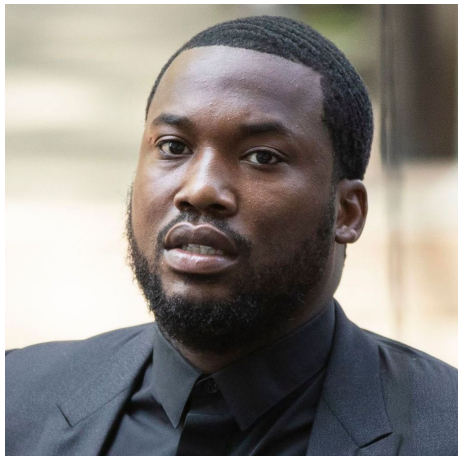
speech

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

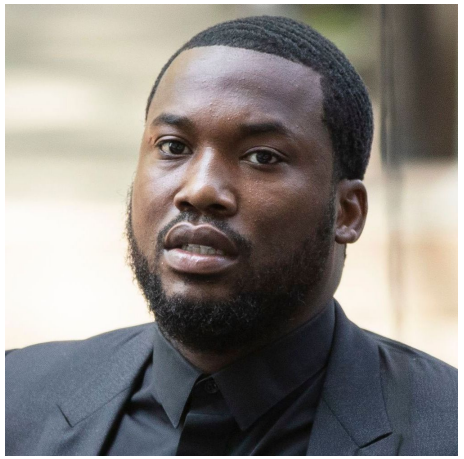
speech

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

Discourse

The study of words' meaningful sub-components

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

ese

Morphology

phonology

phonetics

speech

orthography

text

Multiple levels* to a single word

- Many ways to express the same meaning

Lexical analysis; normalize and disambiguate words

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

these

Lexemes

Morphology

phonology

orthography

phonetics

text

speech

Multiple levels* to a single word

Discourse

Pragmatics

Semantics

Syntax

- 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

pho

phonetics

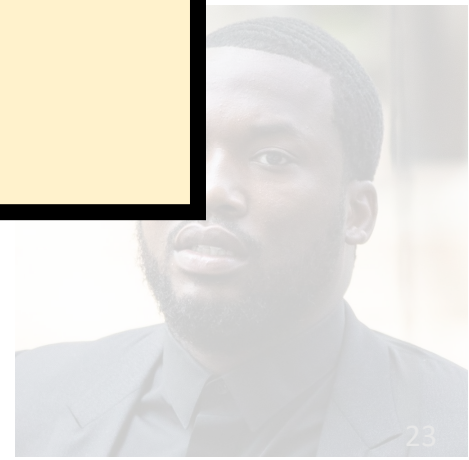
speech

text

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

Determines meaning

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



Multiple levels* to a single word

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

Discourse

Pragmatics

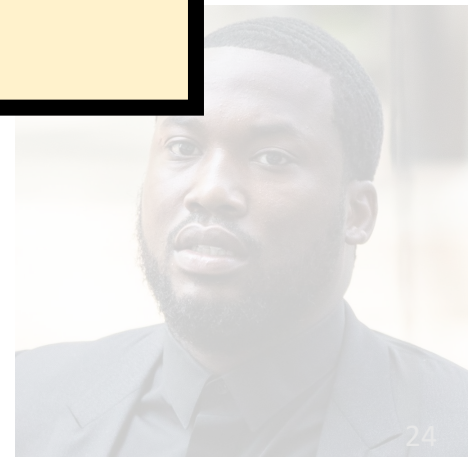
Understands how context affects meaning

(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



Multiple levels* to a single word

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

Discourse

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

phonetics

speech

orthography

text

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.

regardless of context

Token-based

(aka contextualized word representations):

a distinct embedding for every occurrence of every word, completely dependent on its context.

将(单词等)置于上下文中研究

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)

(common sense reasoning)

Slide adapted from or inspired by Richard Socher

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"

Documentation FAQ

en teach

An English term in ConceptNet 5.8

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 аԥцара (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 supervised labels; treat it as a self-supervised task

How

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

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 in every state-of-the-art system.

"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):

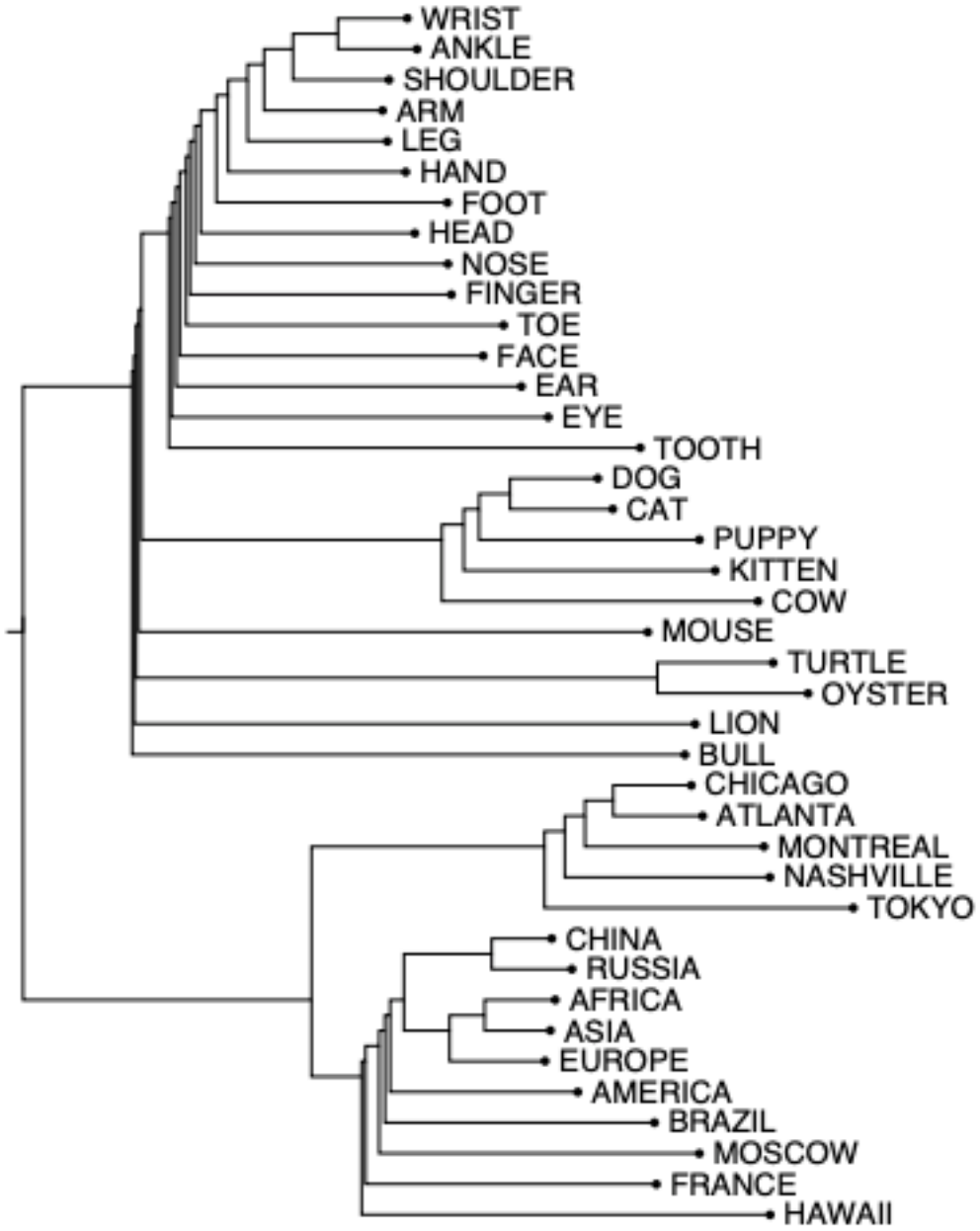
"I like data science. I like computer science. I love data."

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):

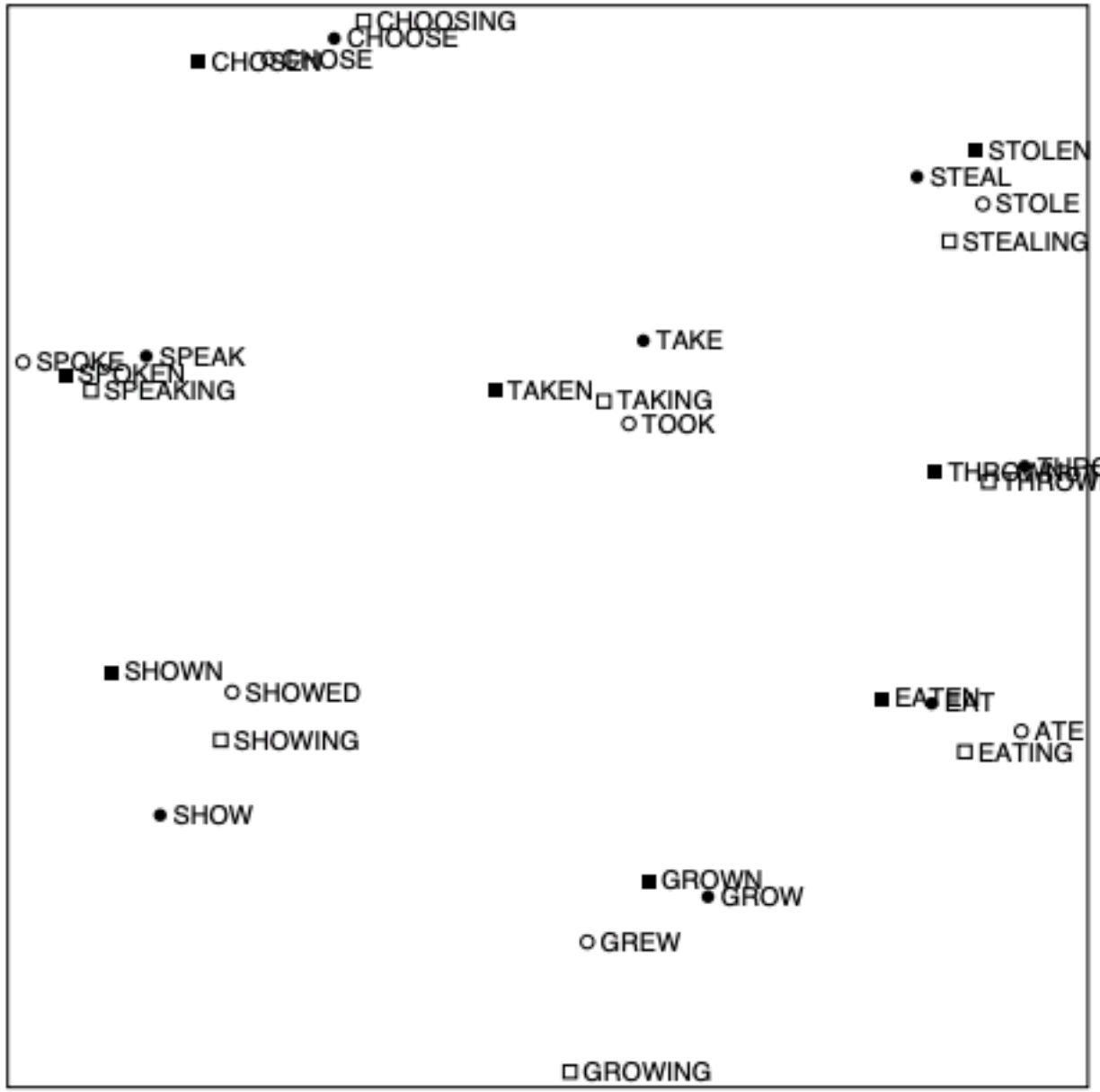
Even these count-based + SVD models can yield interesting results



Slide adapted from or inspired by Richard Socher

Count-based (Distributional Semantic Models):

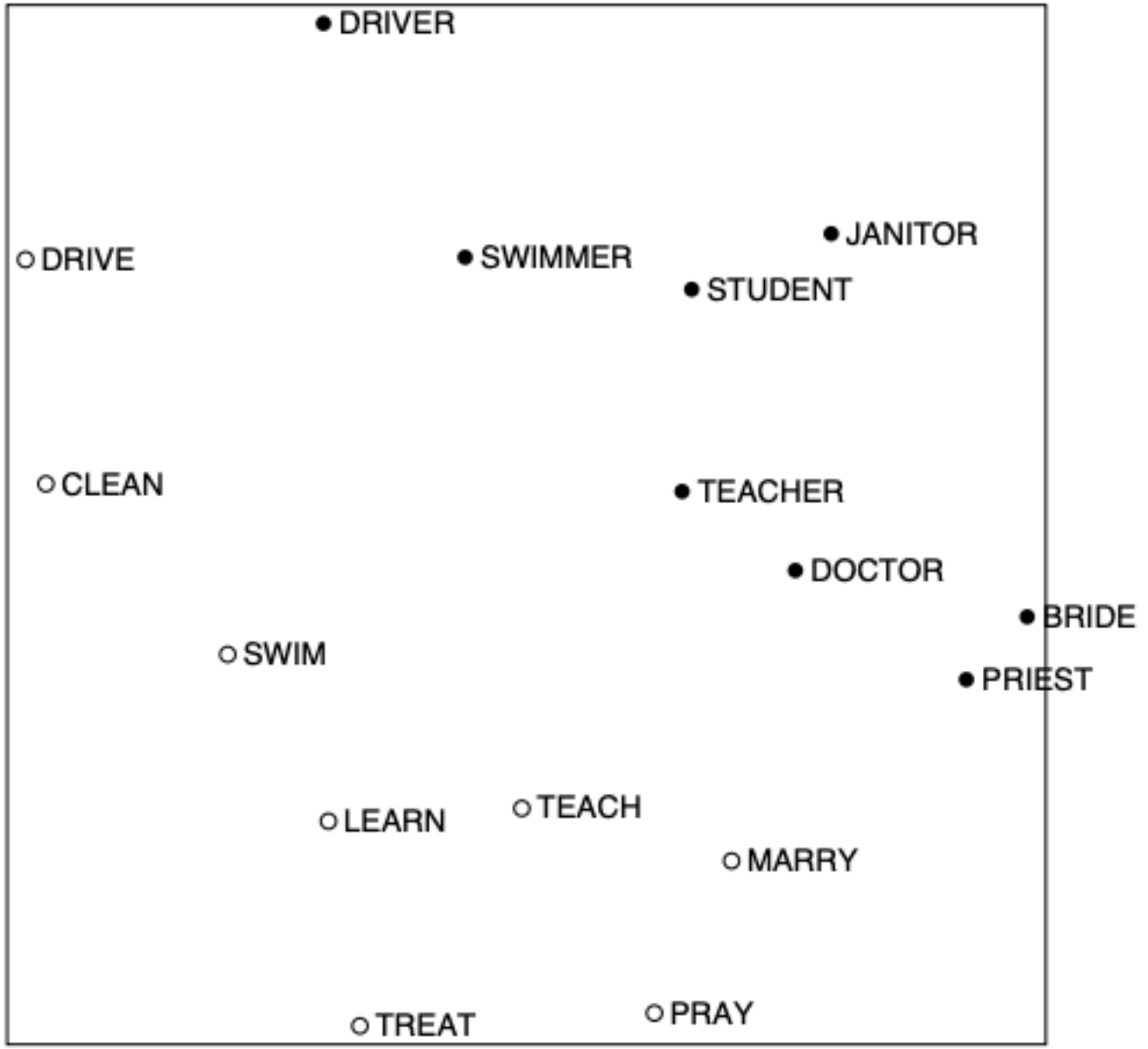
Even these count-based + SVD models can yield interesting results



Slide adapted from or inspired by Richard Socher

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

Slide adapted from or inspired by Richard Socher

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):

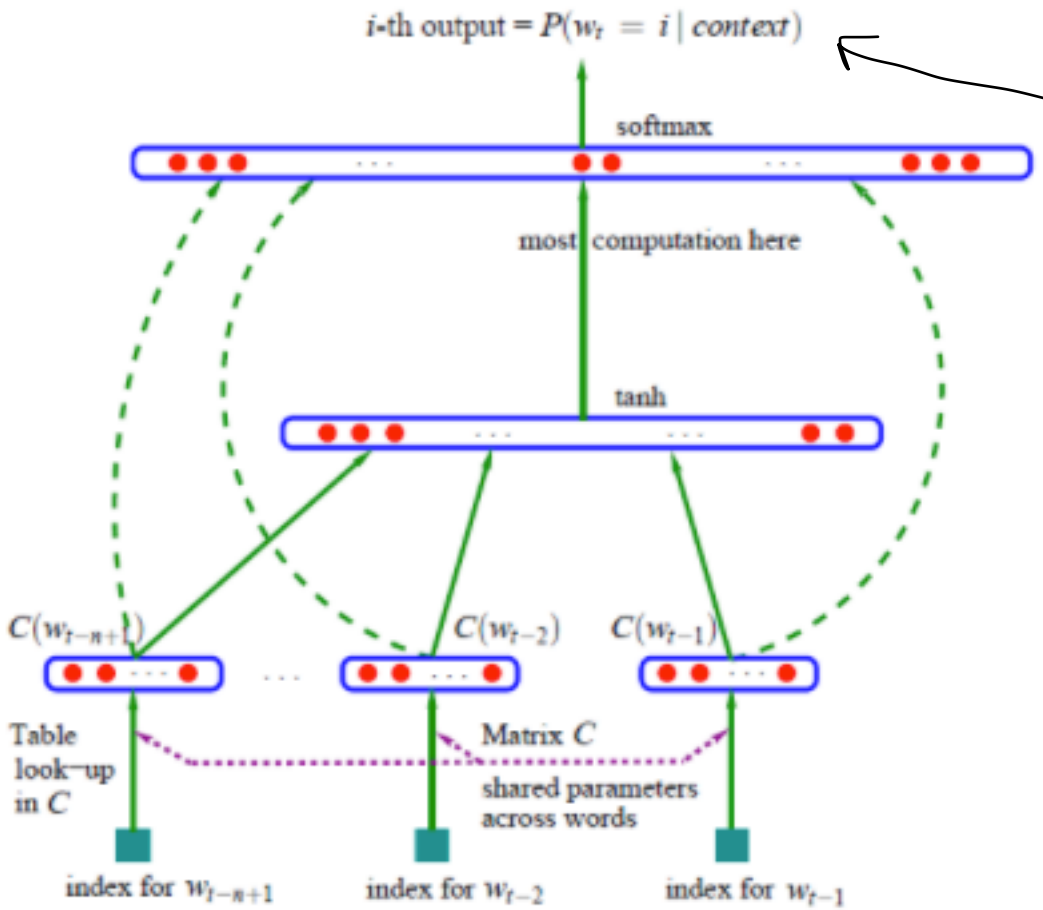


Figure 2: Classic neural language model (Bengio et al., 2003)

- Window of context for input

只能看到前文的内容，预测后文的内容

autoregressive prediction

language model

predict the context given the word

predict the word given context

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

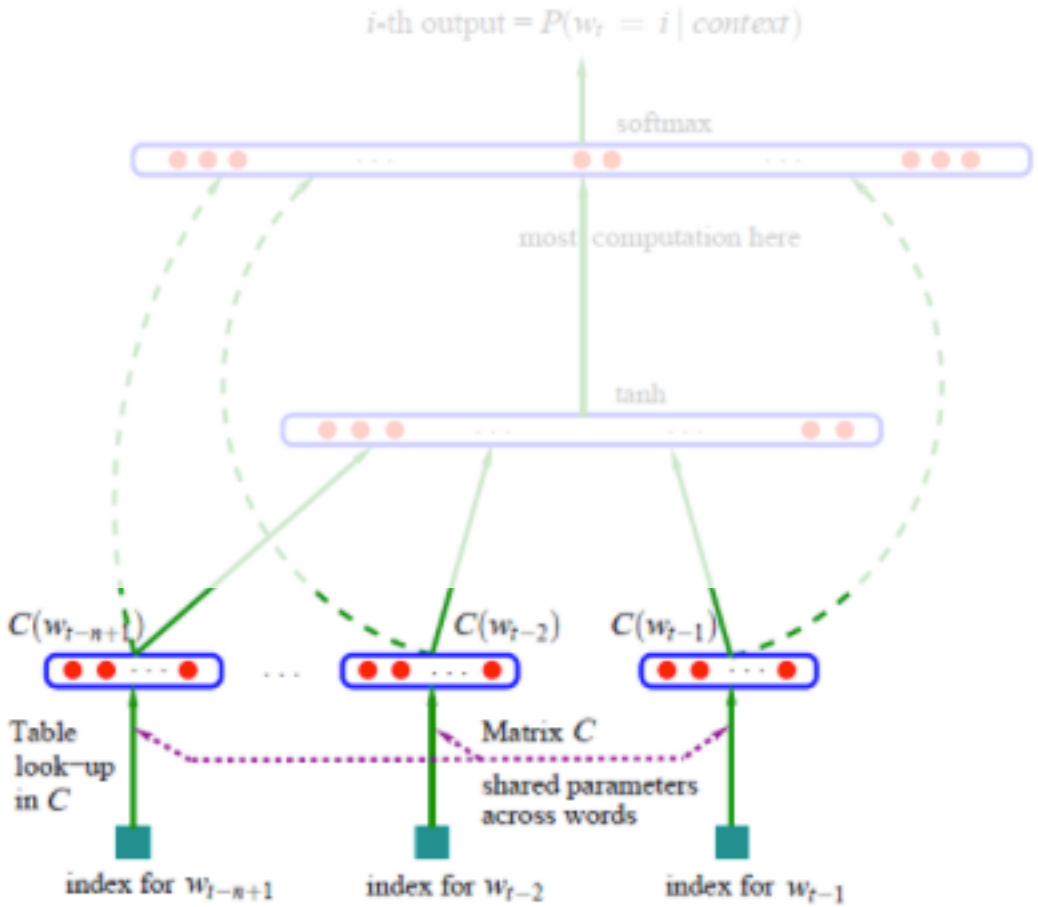


Figure 2: Classic neural language model (Bengio et al., 2003)

- 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):

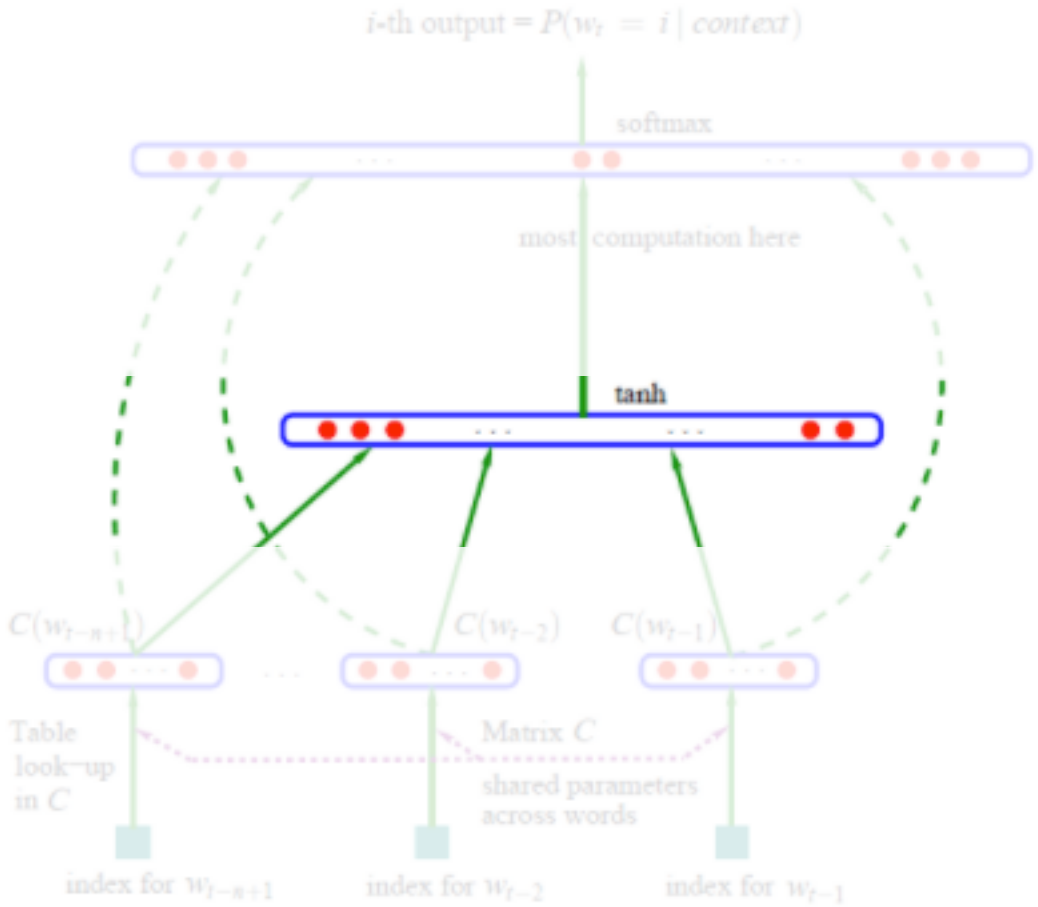


Figure 2: Classic neural language model (Bengio et al., 2003)

- 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):

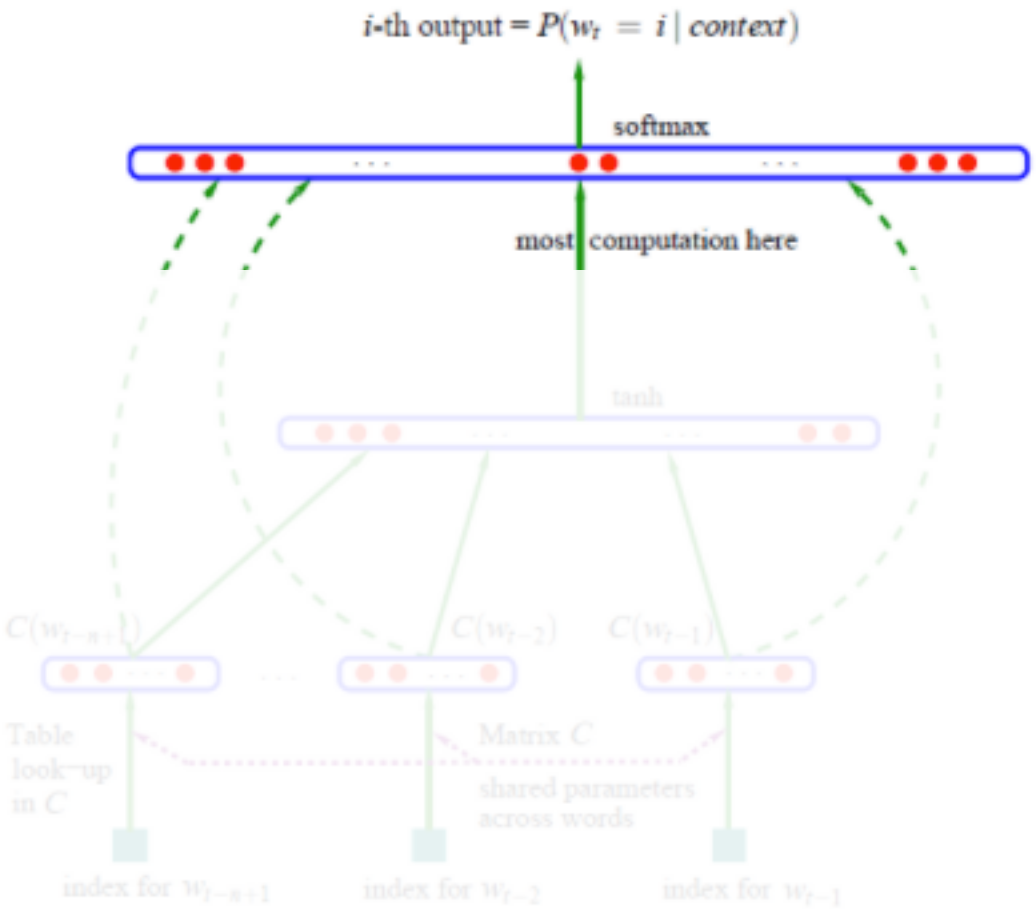


Figure 2: Classic neural language model (Bengio et al., 2003)

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

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

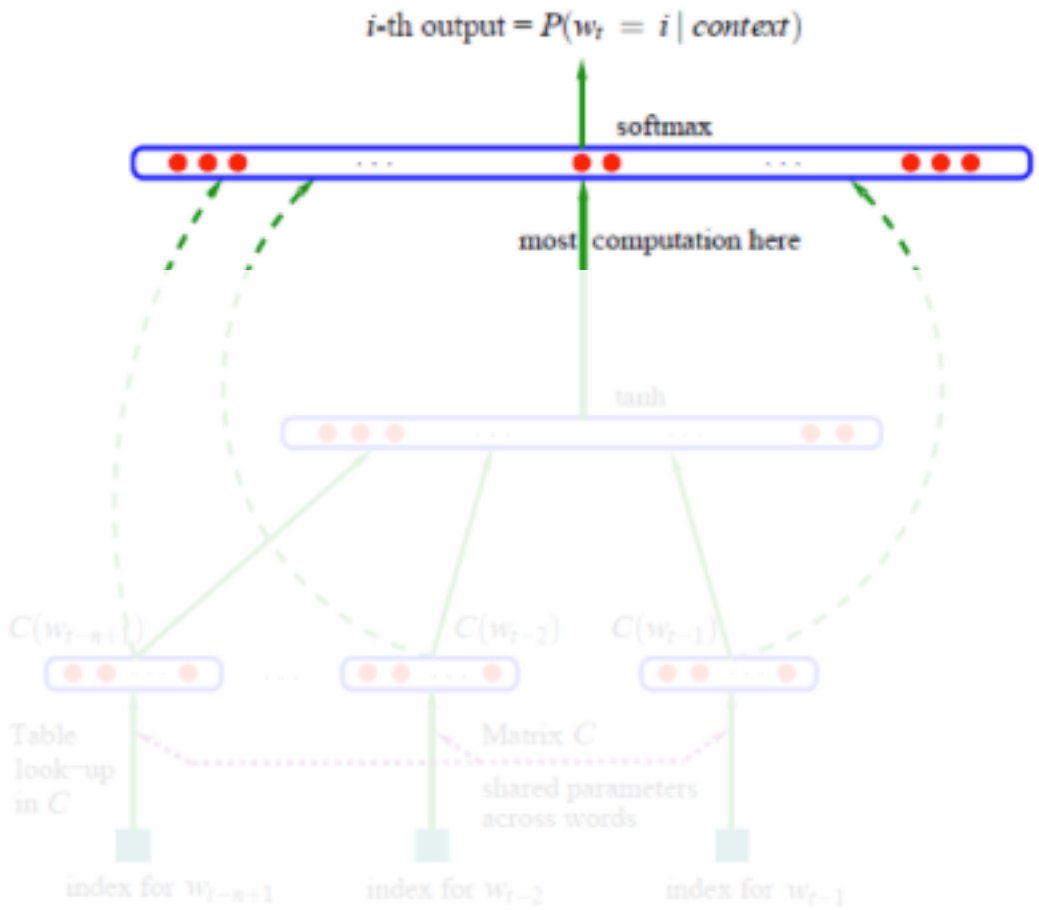


Figure 2: Classic neural language model (Bengio et al., 2003)

- 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)

either predict the center word / context

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)

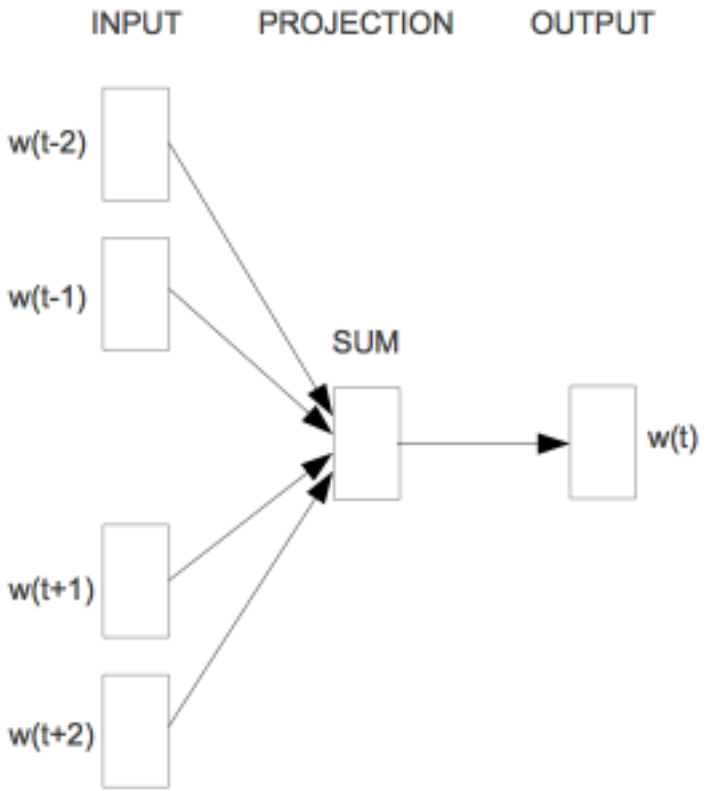


Figure 4: Continuous bag-of-words (Mikolov et al., 2013)

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.

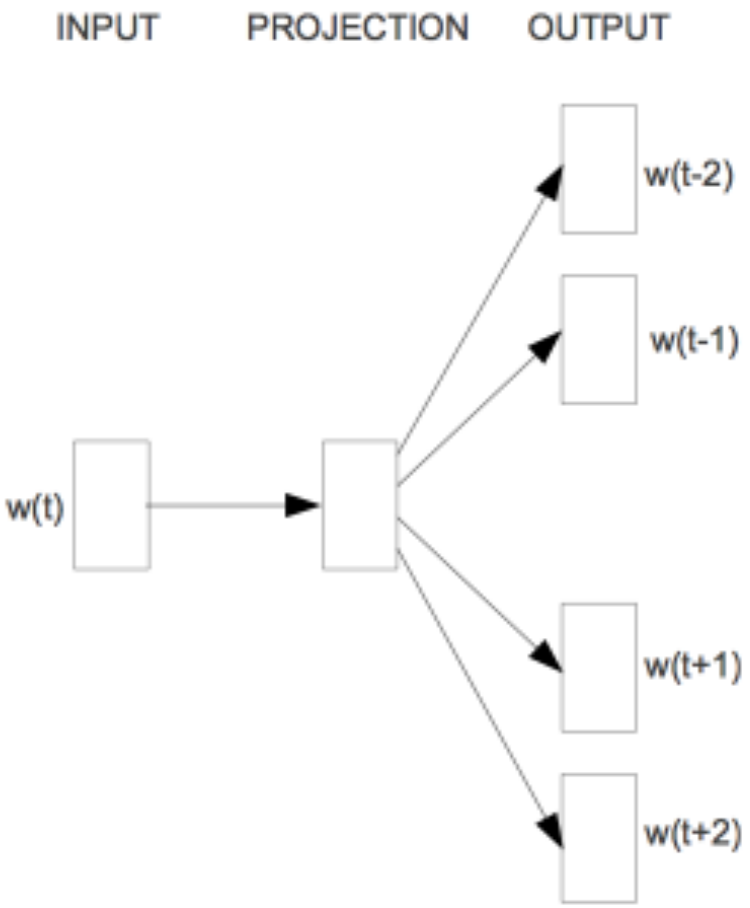


Figure 5: Skip-gram (Mikolov et al., 2013)

word2vec (predictive, not count-based DSMs):

very clever idea of giving which negative samples to this model (how many)

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.

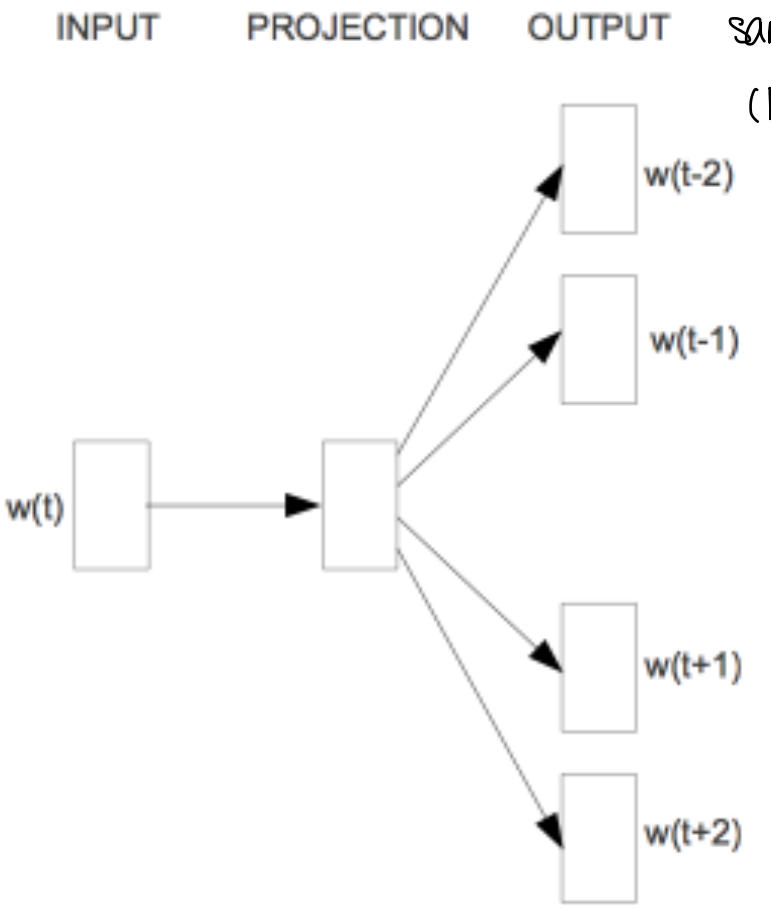


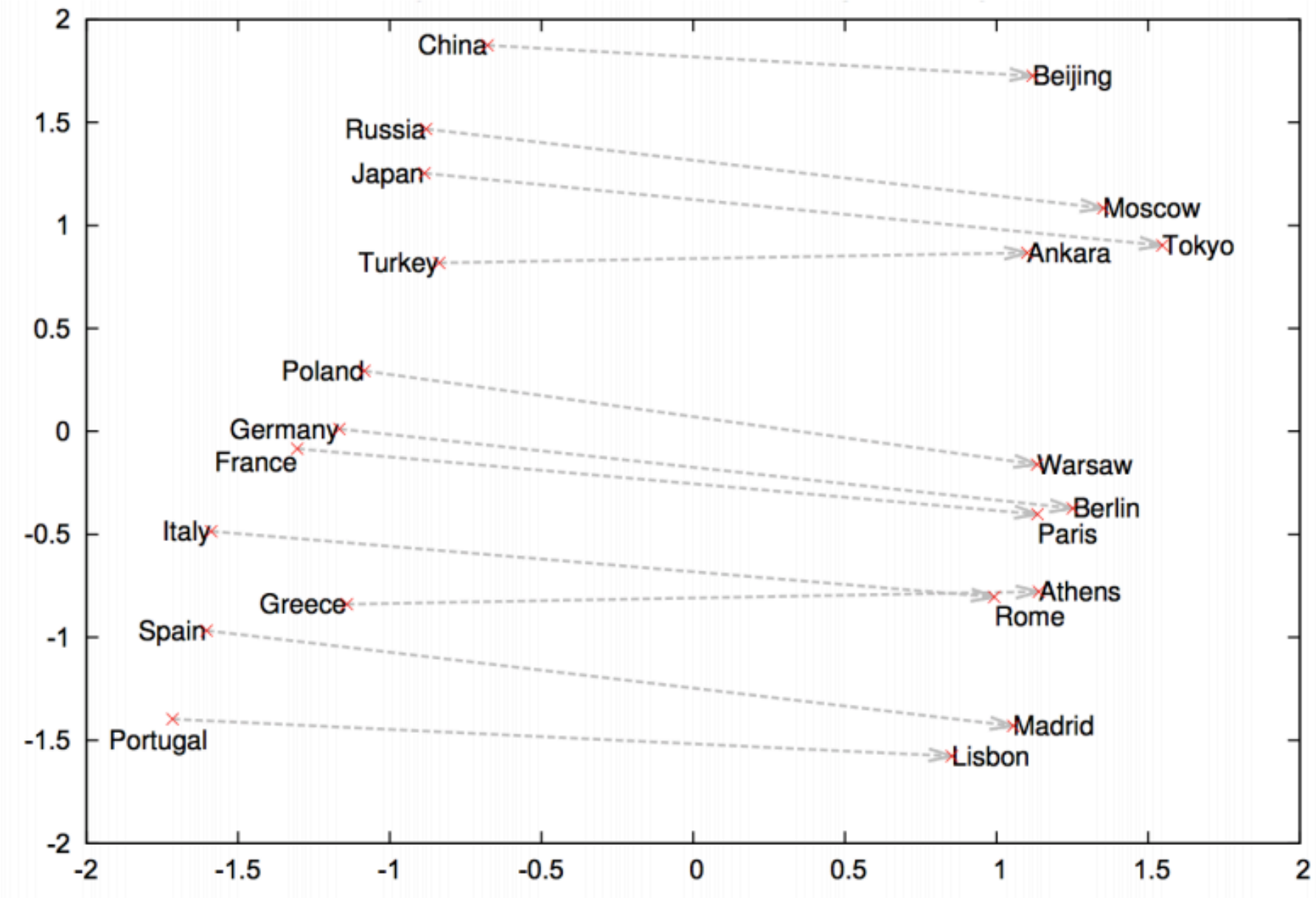
Figure 5: Skip-gram (Mikolov et al., 2013)

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)

how two of the embeddings are related



word2vec (examples of its embeddings)

Incredible finding!!!

(vector)

king − man + woman ~= queen

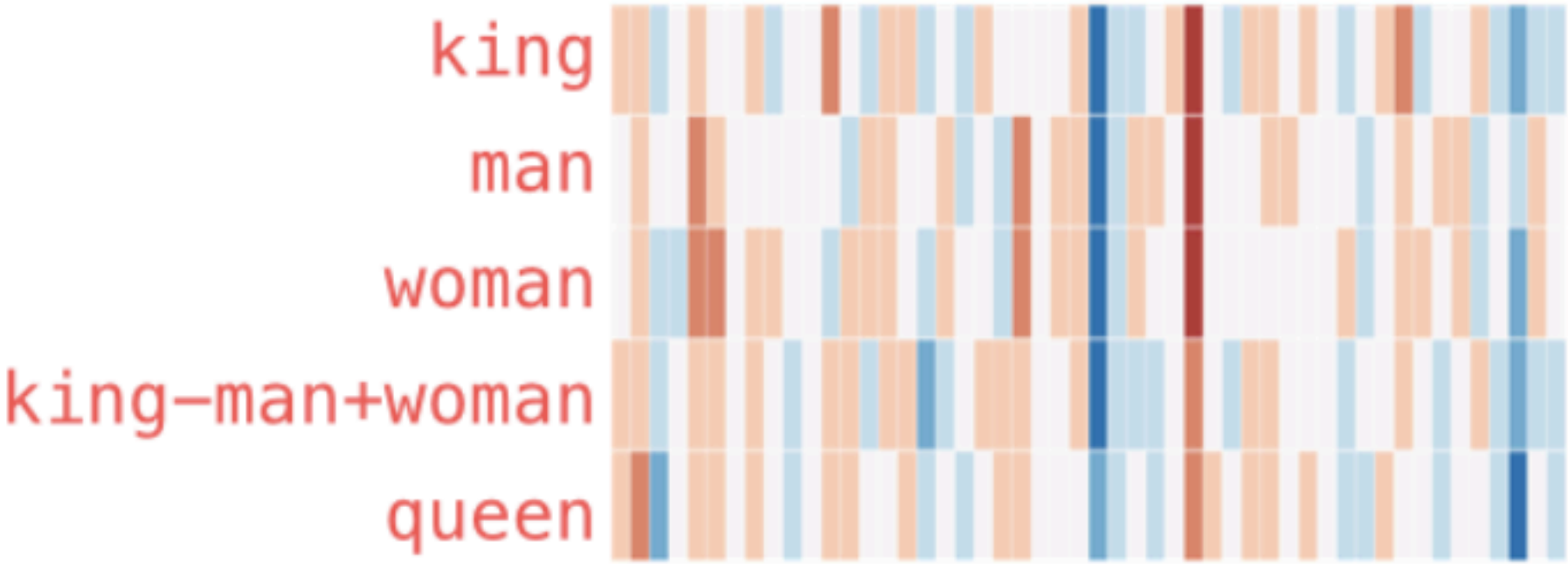


Photo credit: Jay Alammar @ https://jalammar.github.io/illustrated-word2vec/

4、word2vec和NNLM对比有什么区别？（word2vec vs NNLM）

1）其本质都可以看作是语言模型；

2）词向量只不过NNLM一个产物，word2vec虽然其本质也是语言模型，但是其专注于词向量本身，因此做了许多优化来提高计算效率：

与NNLM相比，词向量直接sum，不再拼接，并舍弃隐层；
考虑到sofmax归一化需要遍历整个词汇表，采用hierarchical softmax 和negative sampling进行优化，hierarchical softmax 实质上生成一颗带权路径最小的哈夫曼树，让高频词搜索路劲变小；negative sampling更为直接，实质上对每一个样本中每一个词都进行负例采样；

2）word2vec vs glove

word2vec是局部语料库训练的，其特征提取是基于滑窗的；而glove的滑窗是为了构建co-occurance matrix，是基于全局语料的，可见glove需要事先统计共现概率；因此，word2vec可以进行在线学习，glove则需要统计固定语料信息。
word2vec是无监督学习，同样由于不需要人工标注；glove通常被认为是无监督学习，但实际上glove还是有label的，即共现次数[~~公式~~]。$\log(X_{ij})$
word2vec损失函数实质上是带权重的交叉熵，权重固定；glove的损失函数是最小平方损失函数，权重可以做映射变换。
总体来看，glove可以被看作是更换了目标函数和权重函数的全局word2vec。

不经过优化的CBOW和Skip-gram中 ,在每个样本中每个词的训练过程都要遍历整个词汇表，也就是都需要经过softmax归一化，计算误差向量和梯度以更新两个词向量矩阵（这两个词向量矩阵实际上就是最终的词向量，可认为初始化不一样），当语料库规模变大、词汇表增长时，训练变得不切实际。为了解决这个问题，word2vec支持两种优化方法：hierarchical softmax 和negative sampling。

hierarchical softmax 使用一颗二叉树表示词汇表中的单词，每个单词都作为二叉树的叶子节点。对于一个大小为V的词汇表，其对应的二叉树包含V-1非叶子节点。假如每个非叶子节点向左转标记为1，向右转标记为0，那么每个单词都具有唯一的从根节点到达该叶子节点的由｛0 1｝组成的代号（实际上为哈夫曼编码，为哈夫曼树，是带权路径长度最短的树，哈夫曼树保证了词频高的单词的路径短，词频相对低的单词的路径长，这种编码方式很大程度减少了计算量）。

CBOW中的目标函数是使条件概率 ~~[公式]~~ 最大化 $P(w|content(w))$

Skip-gram中的目标函数是使条件概率 ~~[公式]~~ 最大化 $P(content(w)|w)$

negative sampling是一种不同于hierarchical softmax的优化策略，相比于hierarchical softmax，negative sampling的想法更直接——为每个训练实例都提供负例。

负采样算法实际上就是一个带权采样过程，负例的选择机制是和单词词频联系起来的。

具体做法是以 N+1 个点对区间 [0,1] 做非等距切分，并引入的一个在区间 [0,1] 上的 M 等距切分，其中 M >> N。源码中取 M = 10^8。然后对两个切分做投影，得到映射关系：采样时，每次生成一个 [1, M-1] 之间的整数 i，则 Table(i) 就对应一个样本；当采样到正例时，跳过（拒绝采样）。

Glove：

（1）根据语料库构建一个共现矩阵，矩阵中的每一个元素 X_{ij} 代表单词 i 和上下文单词 j 在特定大小的上下文窗口内共同出现的次数。

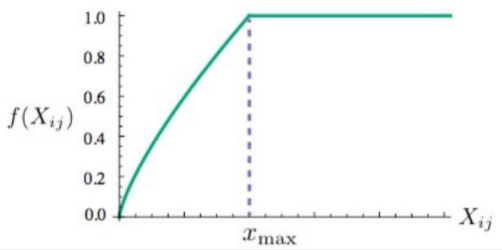
（2）构建词向量（Word Vector）和共现矩阵之间的近似关系，其目标函数为：

$$J=\sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j+b_i+b_j-\log X_{ij}\right)^2$$

这个loss function的基本形式就是最简单的mean square loss，只不过在此基础上加了一个权重函数 $f(x_{ij})$：

$$f(x)=\begin{cases}(x/x_{\max})^\alpha & \text{if } x<x_{\max}\\ 1 & \text{otherwise.}\end{cases}$$

根据实验发现 $x_{\max}$ 的值对结果的影响并不是很大，原作者采用了 $x_{\max}=100$ 。而 $\alpha=3/4$ 时的结果要比 $\alpha=1$ 时要更好。下面是 $\alpha=3/4$ 时 $f(x)$ 的函数图象，可以看出对于较小的 X_{ij} ，权值也较小。这个函数图像如下所示：



2、GloVe的训练过程是怎样的?

1. 实质上还是监督学习：虽然glove不需要人工标注为无监督学习，但实质还是有label就是 $log(X_{ij})$ 。
2. 向量 w 和 $\tilde{w}$ 为学习参数，本质上与监督学习的训练方法一样，采用了AdaGrad的梯度下降算法，对矩阵 X 中的所有非零元素进行随机采样，学习曲率（learning rate[9]）设为0.05，在vector size小于300的情况下迭代了50次，其他大小的vectors上迭代了100次，直至收敛。
3. 最终学习得到的是两个词向量是 $\tilde{w}$ 和 w ，因为 X 是对称的（symmetric），所以从原理上讲 $\tilde{w}$ 和 w ，是也是对称的，他们唯一的区别是初始化的值不一样，而导致最终的值不一样。所以这两者其实是等价的，都可以当成最终的结果来使用。但是为了提高鲁棒性，我们最终会选择两者之和 $w+\tilde{w}$ 作为最终的vector（两者的初始化不同相当于加了不同的随机噪声，所以能提高鲁棒性）。

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

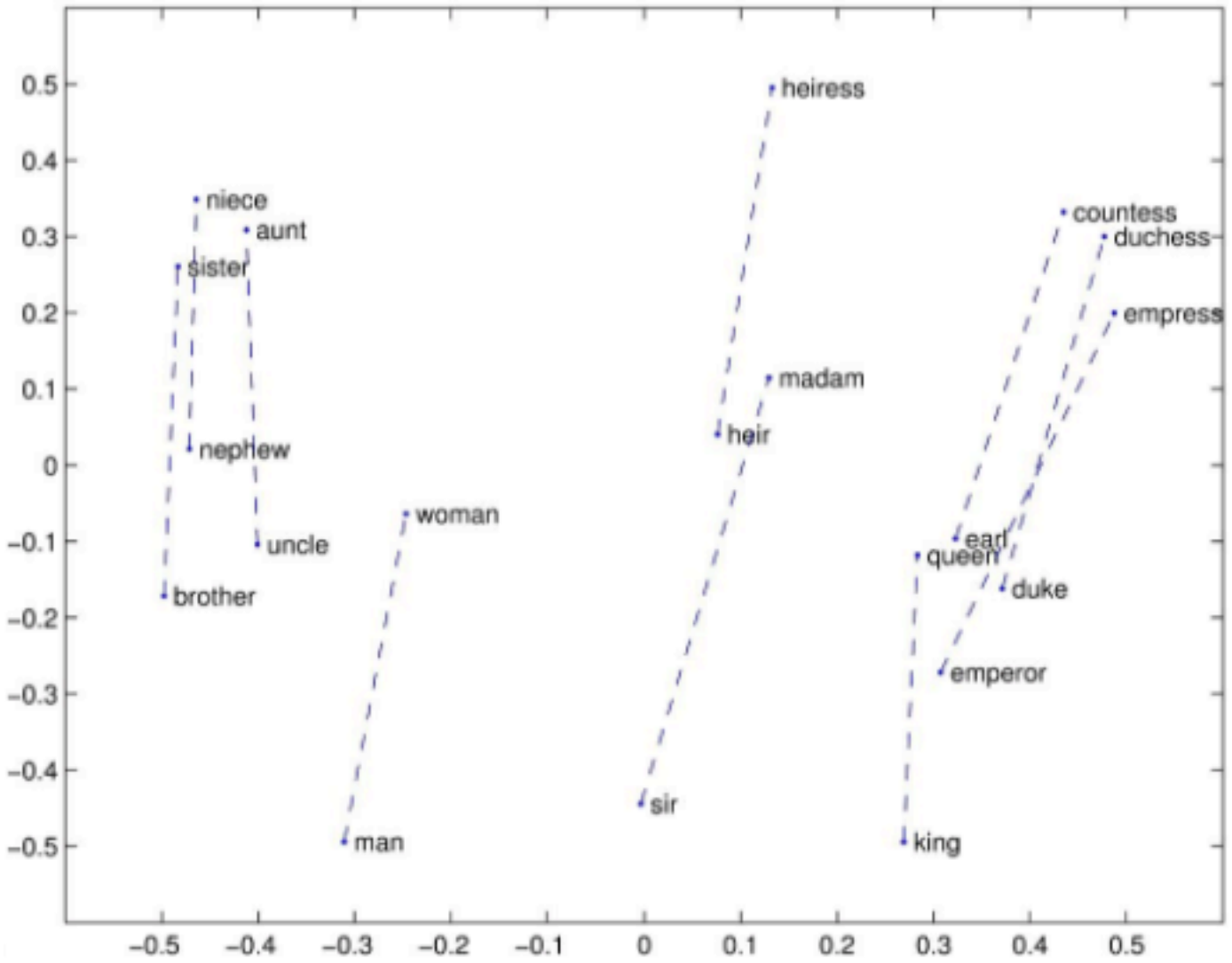
	x = solid	x = gas	x = water	x = random
$P(x\|\text{ice})$	large	small	large	small
$P(x\|\text{steam})$	small	large	large	small
$\frac{P(x\|\text{ice})}{P(x\|\text{steam})}$	large	small	~1	~1

GloVe (predictive, not count-based DSMs):

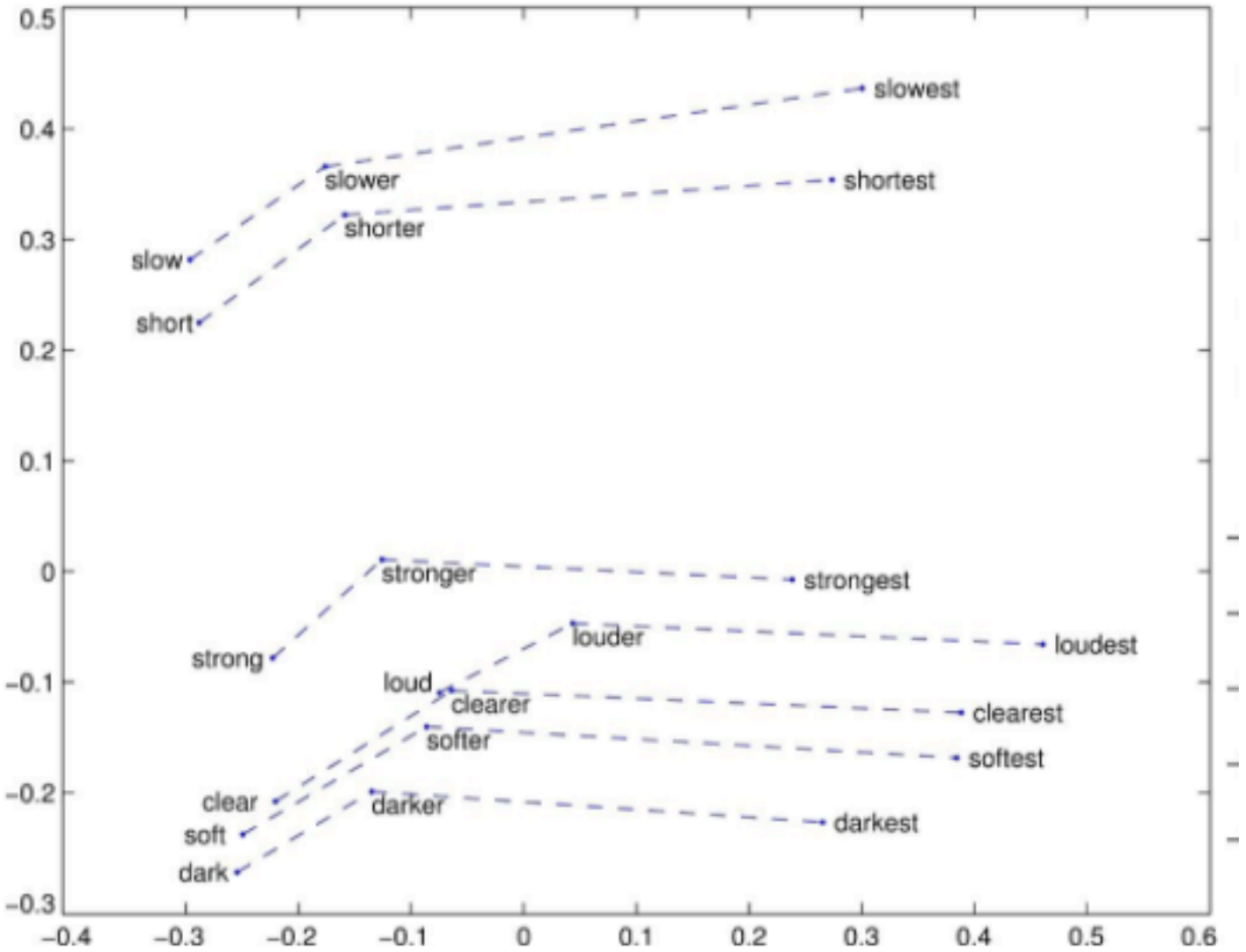
Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	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):



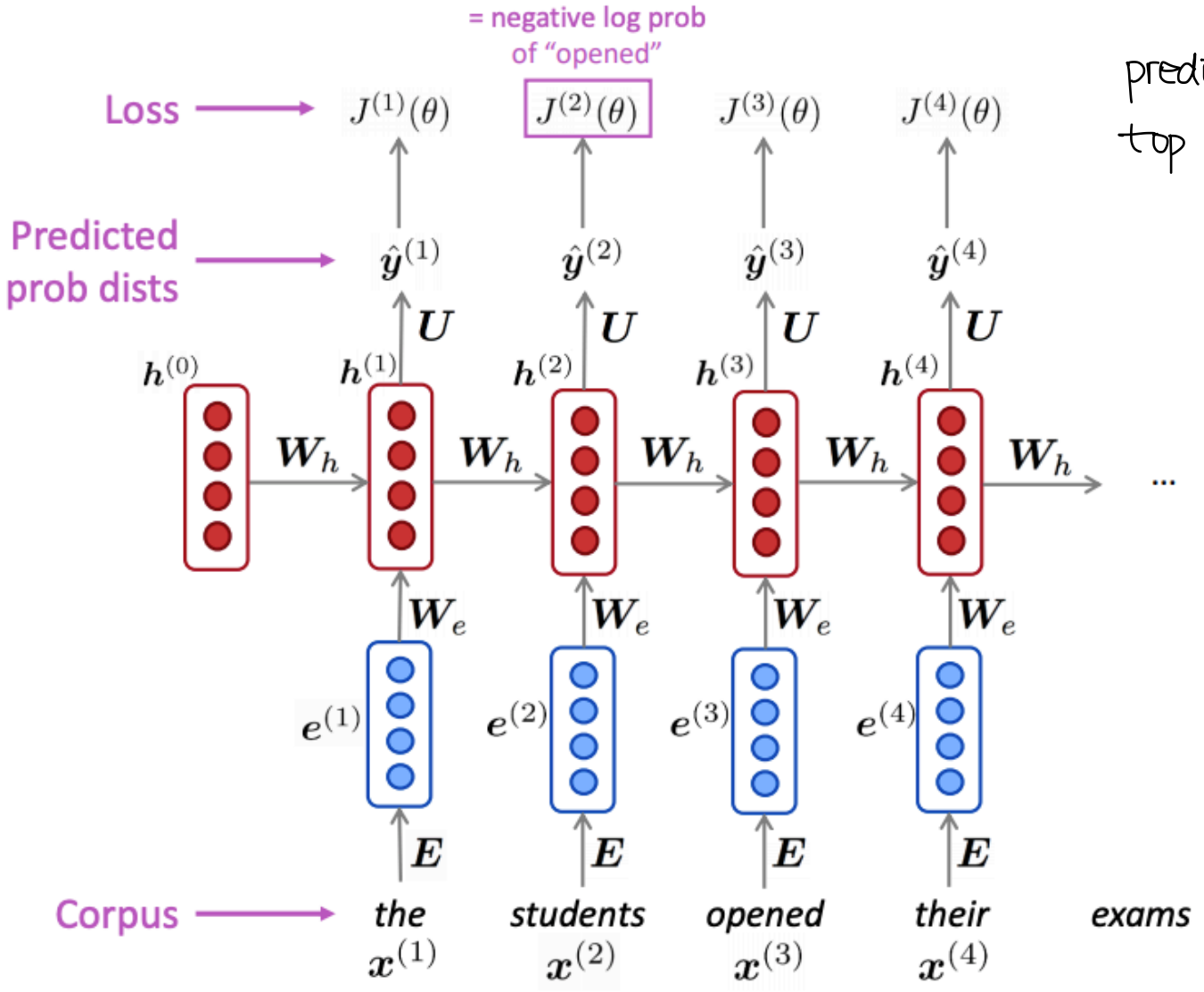
TAKEAWAYS

- 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)



predict the next word

top layer ⇒ classification task

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

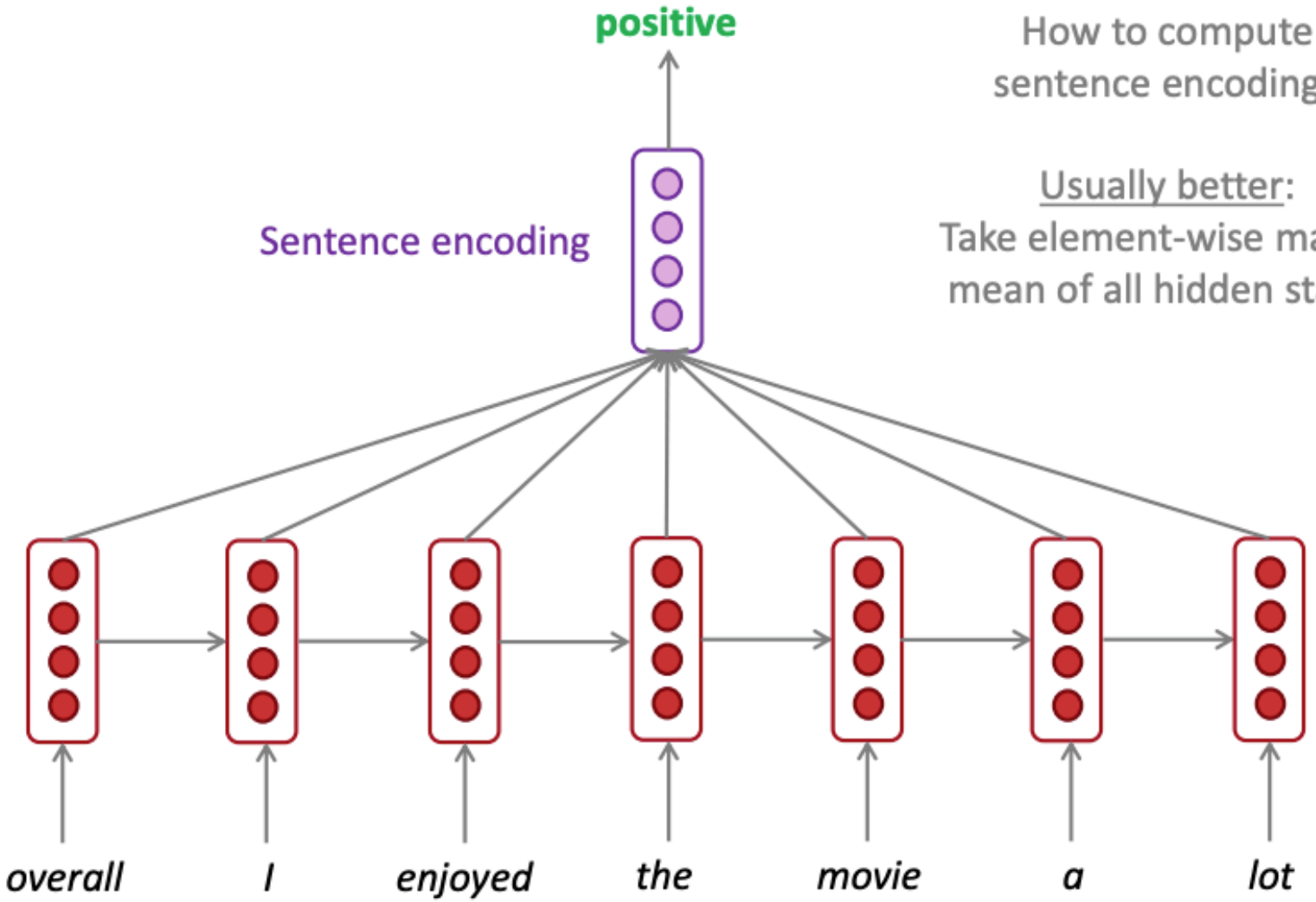


Photo credit: Abigail See

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.