Language Models

CS288
UC Berkeley

Recap: N-Gram Models

- Use chain rule to generate words left-to-right

$$P(w_1 \dots w_n) = \prod_i P(w_i | w_1 \dots w_{i-1})$$

- Can't condition atomically on the entire left context

P(??? | The computer I had put into the machine room on the fifth floor just)

- N-gram models make a Markov assumption

$$P(w_1 \dots w_n) = \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

$$P(\text{please close the door}) = \quad P(\text{please}|\text{START})P(\text{close}|\text{please}) \dots P(\text{STOP}|door)$$

Empirical N-Grams

- Use statistics from data (examples here from Google N-Grams)

Training Counts

198015222 the first
194623024 the same
168504105 the following
158562063 the world
…
14112454 the door

23135851162 the *

$$\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162}$$

$$= 0.0006$$

- This is the maximum likelihood estimate, which needs modification
- N-gram models use such counts to compute probabilities on demand

Increasing N-Gram Order

- Higher orders capture more correlations

Bigram Model

198015222	the first
194623024	the same
168504105	the following
158562063	the world
…	
14112454	the door

23135851162	the *

Trigram Model

197302	close the window
191125	close the door
152500	close the gap
116451	close the thread
87298	close the deal

3785230	close the *

P(door | the) = 0.0006

P(door | close the) = 0.05

N-Gram Models: Challenges

Sparsity

Please close the first door on the left.

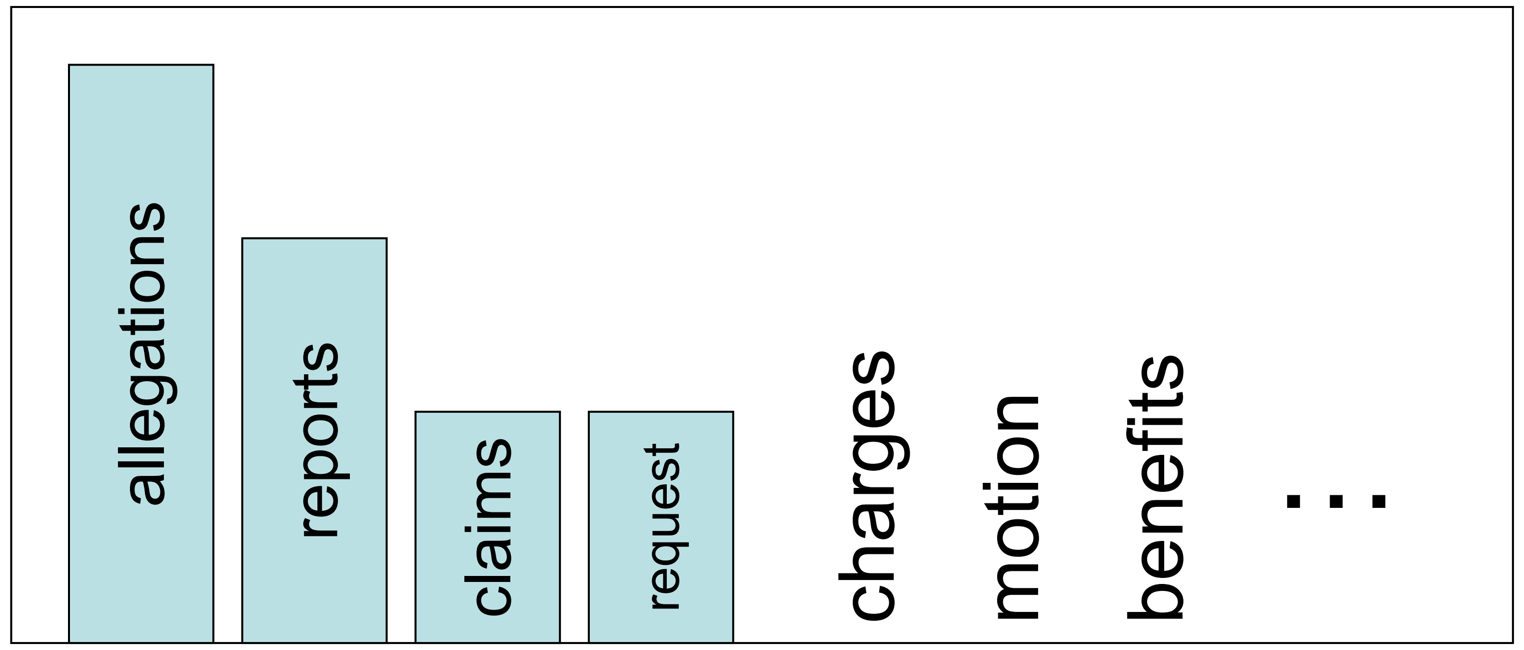
3380 please close the door
1601 please close the window
1164 please close the new
1159 please close the gate
…
0 please close the first

13951 please close the *

Smoothing

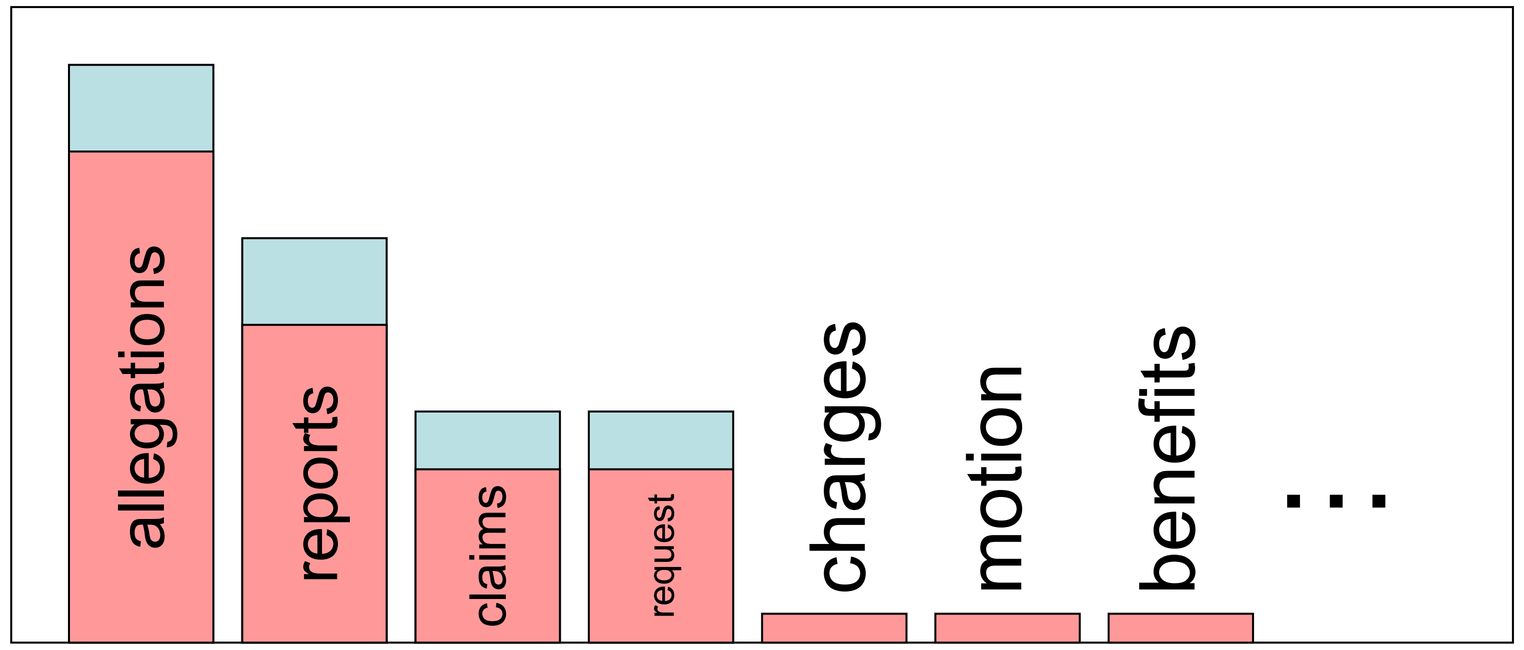
- We often want to make estimates from sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



- Smoothing flattens spiky distributions so they generalize better:

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



- Very important all over NLP, but easy to do badly

Back-off

Please close the first door on the left.

4-Gram	3-Gram	2-Gram
3380 please close the door	197302 close the window	198015222 the first
1601 please close the window	191125 close the door	194623024 the same
1164 please close the new	152500 close the gap	168504105 the following
1159 please close the gate	116451 close the thread	158562063 the world
…	…	…
0 please close the first	8662 close the first	…
-----------------	-----------------	-----------------
13951 please close the *	3785230 close the *	23135851162 the *
0.0	0.002	0.009

Specific but Sparse ⟷ Dense but General

$$\lambda \hat{P}(w|w_{-1}, w_{-2}) + \lambda' \hat{P}(w|w_{-1}) + \lambda'' \hat{P}(w)$$

Discounting

- Observation: N-grams occur more in training data than they will later

Empirical Bigram Counts (Church and Gale, 91)

Count in 22M Words	Future c* (Next 22M)
1	
2	
3	
4	
5	

- Absolute discounting: reduce counts by a small constant, redistribute "shaved" mass to a model of new events

$$P_{\text{ad}}(w|w') = \frac{c(w',w) - d}{c(w')} + \alpha(w')\hat{P}(w)$$

Fertility

- Shannon game: "There was an unexpected _____"

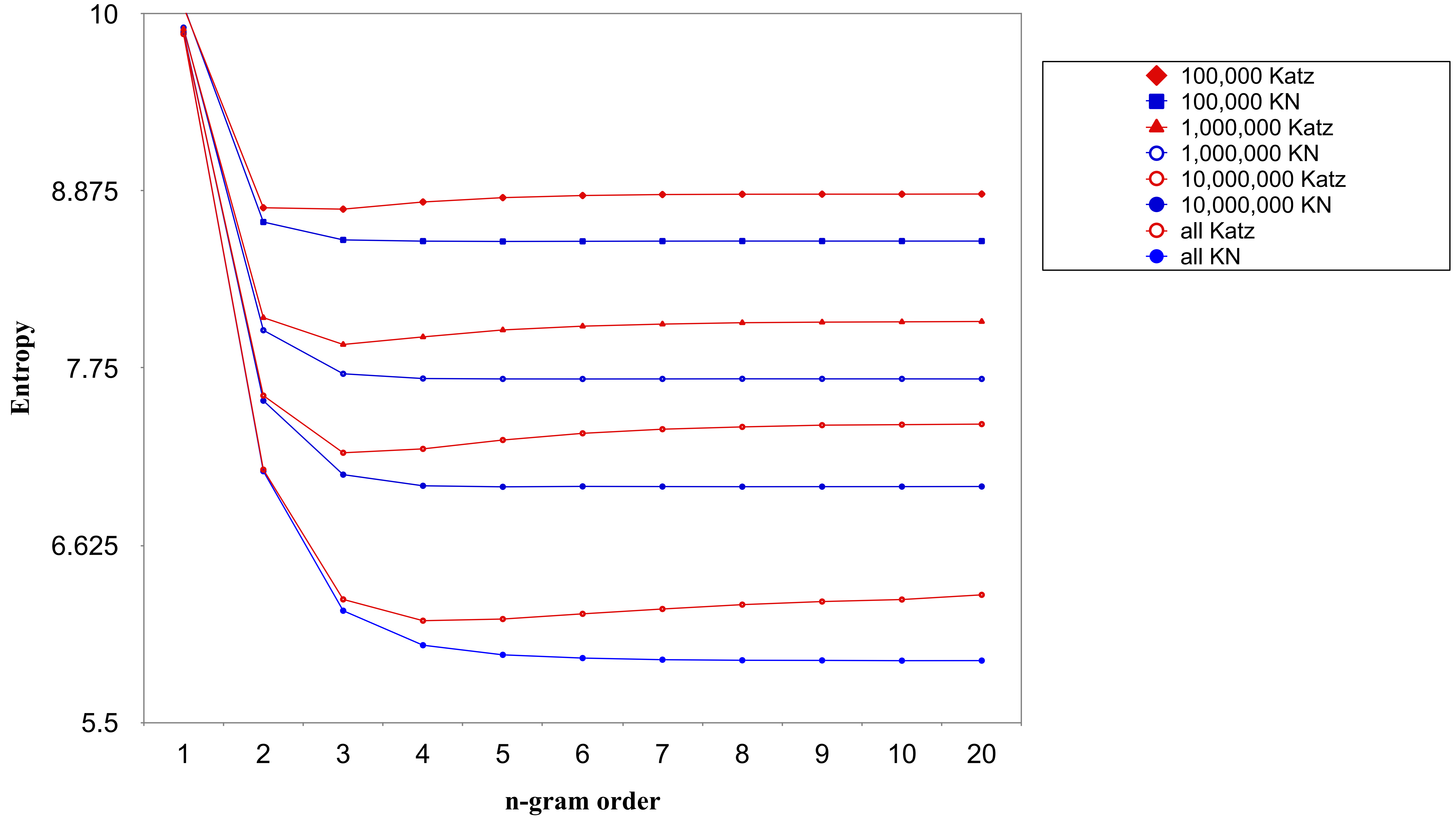
delay?   Francisco?

- Context fertility: number of distinct context types that a word occurs in
  - What is the fertility of "delay"?
  - What is the fertility of "Francisco"?
  - Which is more likely in an arbitrary new context?
- Kneser-Ney smoothing: new events proportional to context fertility, not frequency

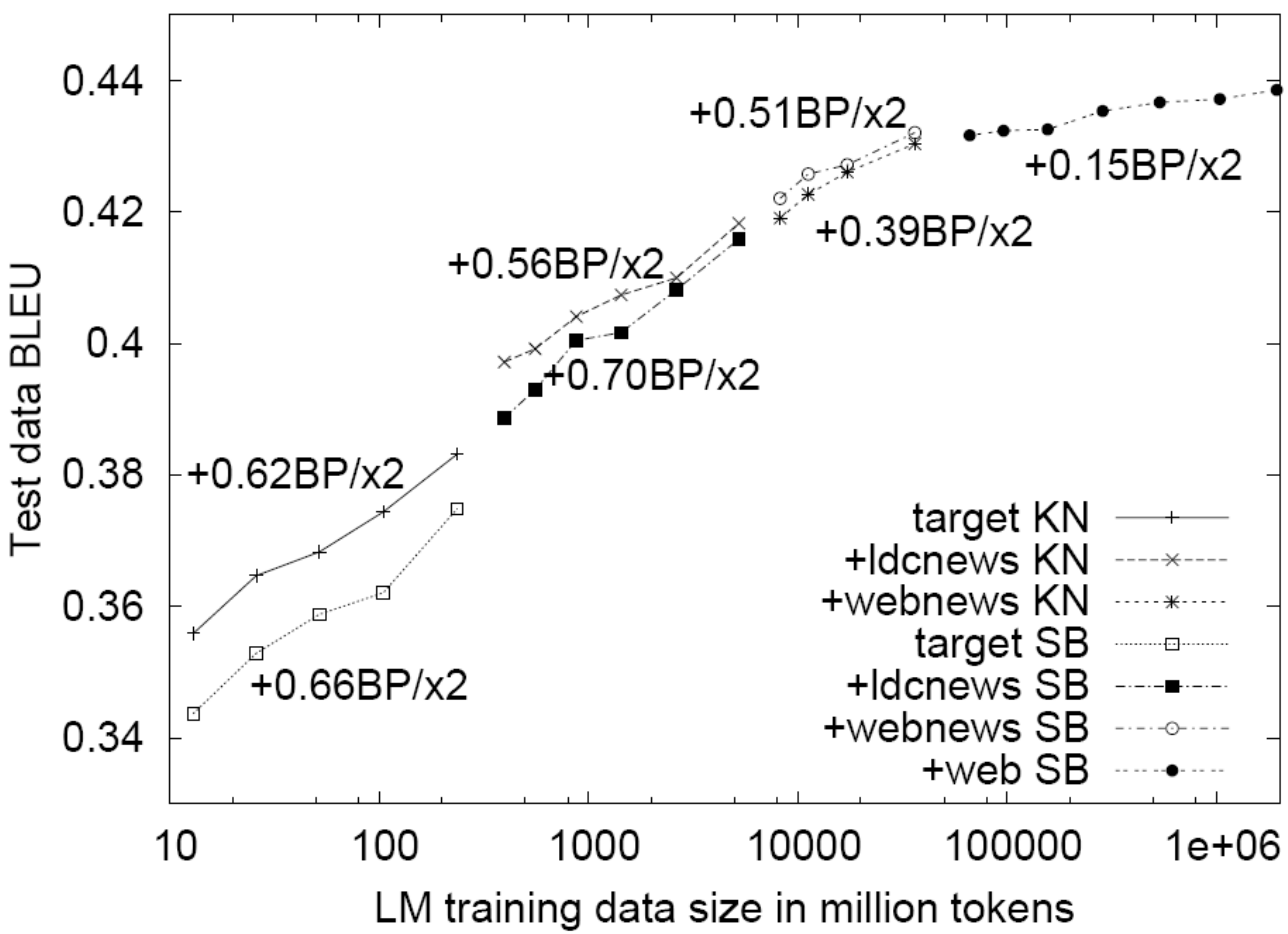
[Kneser & Ney, 1995]

$$P(w) \propto |\{w' : c(w', w) > 0\}|$$

Better Methods?



More Data?



[Brants et al, 2007]

Storage

…	
searching for the best	192593
searching for the right	45805
searching for the cheapest	44965
searching for the perfect	43959
searching for the truth	23165
searching for the “	19086
searching for the most	15512
searching for the latest	12670
searching for the next	10120
searching for the lowest	10080
searching for the name	8402
searching for the finest	8171
…	

Google N-grams

- 14 million < 2^{24} words
- 2 billion < 2^{31} 5-grams
- 770 000 < 2^{20} unique counts
- 4 billion n-grams total

Entirely Unseen Words

- What about totally unseen words?

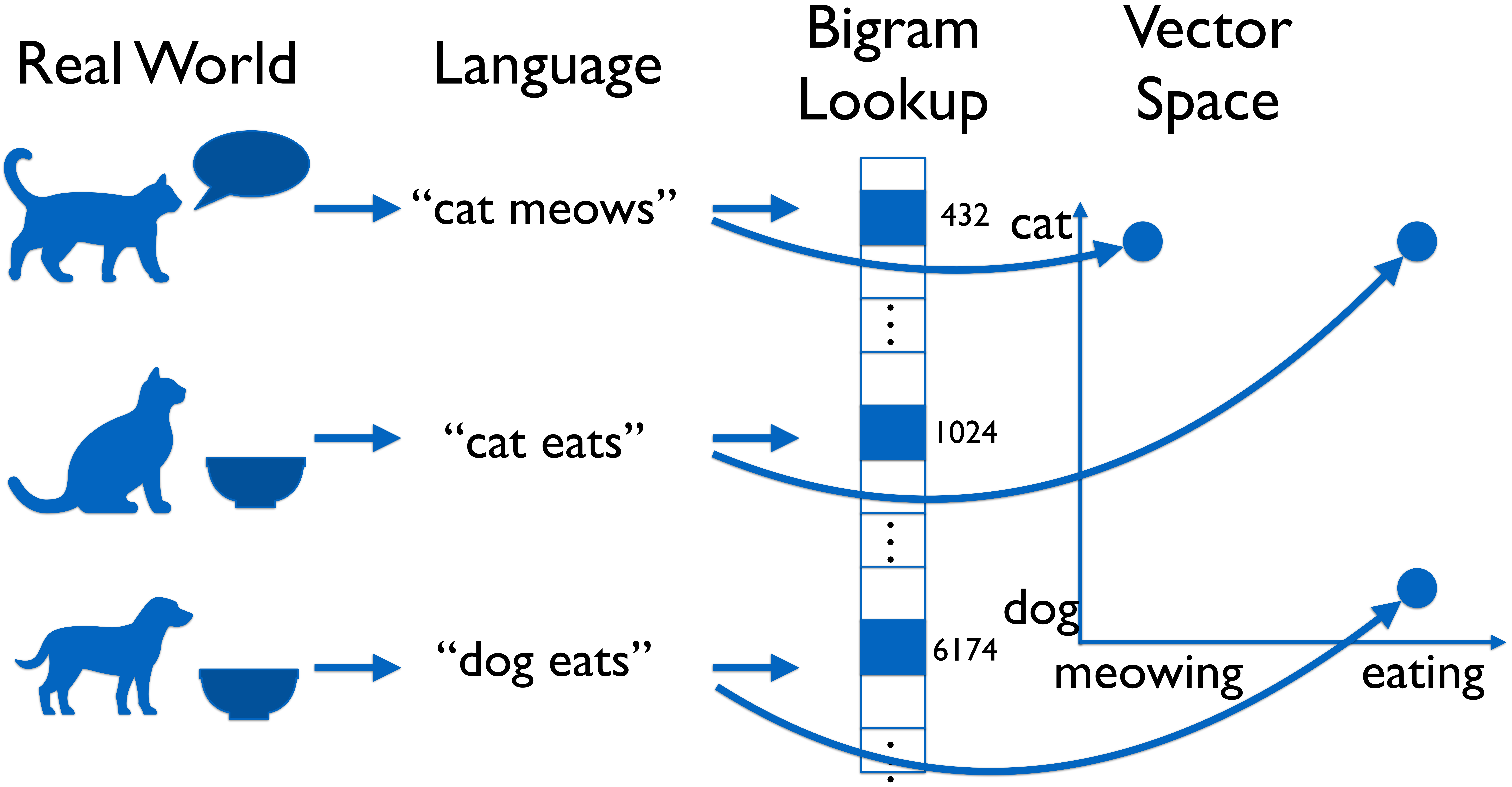
- Classical real world option: systems are actually closed vocabulary
  - ASR systems will only propose words that are in their pronunciation dictionary
  - MT systems will only propose words that are in their phrase tables (modulo special models for numbers, etc)

- Classical theoretical option: build open vocabulary LMs
  - Models over character sequences rather than word sequences
  - N-Grams: back-off needs to go down into a “generate new word” model
  - Typically if you need this, a high-order character model will do

- Modern approach: syllable-sized subword units (more later)

Representation Learning

What is a Representation?

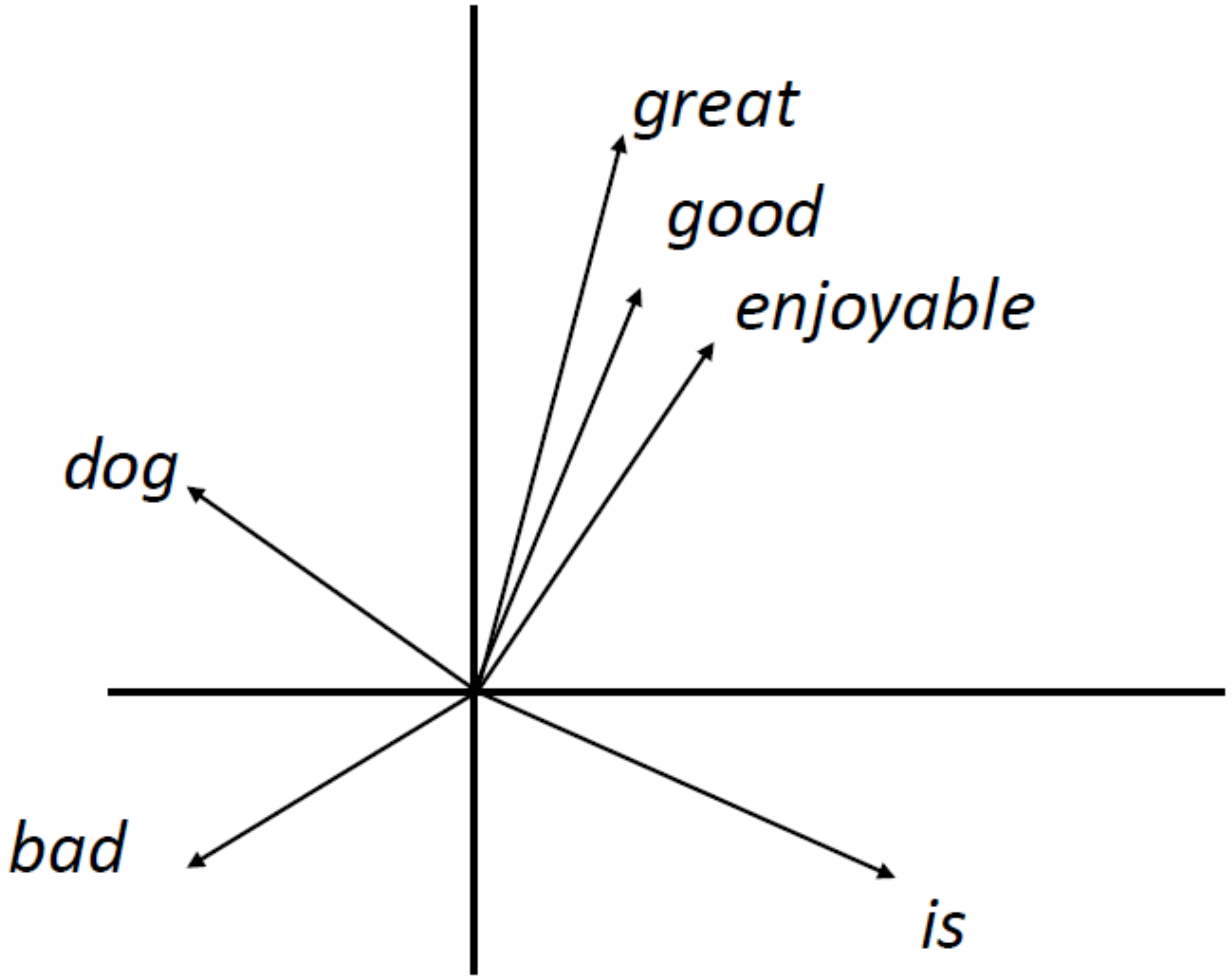


Vector Embeddings

- Embeddings map discrete words (eg |V| = 50k) to continuous vectors (eg d = 100)

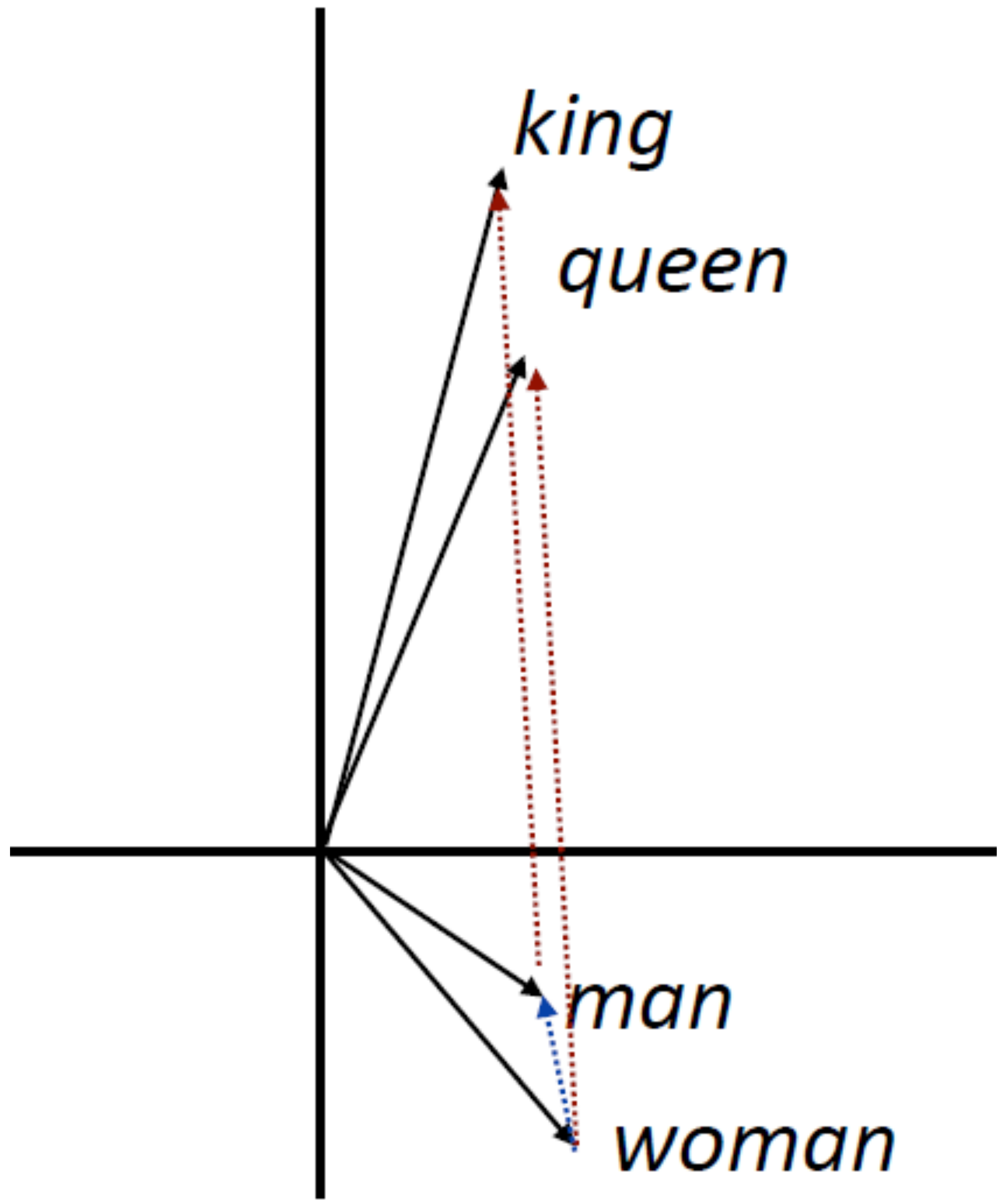
- Why do we care about embeddings?
  - Neural methods want them
  - Nuanced similarity possible; generalize across words

- We hope embeddings will have structure that exposes word correlations (and thereby meanings)



Structure of Embedding Spaces

- How can you fit 50K words into a 64-dimensional hypercube?
- Orthogonality: Can each axis have a global “meaning” (number, gender, animacy, etc)?
- Global structure: Can embeddings have algebraic structure (eg king – man + woman = queen)?



Bias in Embeddings

- Embeddings can capture biases in the data! (Bolukbasi et al 16)

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

- Debiasing methods (as in Bolukbasi et al 16) are an active area of research

What can we Embed?

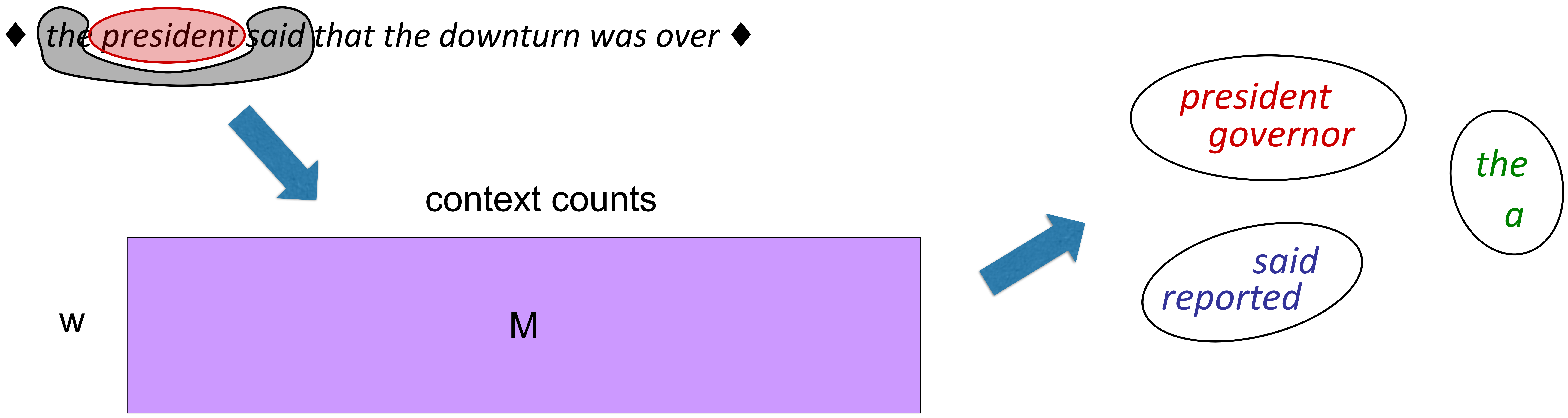
- Subwords
- Words
- N-grams
- Entire sentences
- Entire documents
- Things that aren’t text (e.g., images)

Stuffing Meanings into Vector Spaces?

Cartoon: Greg Durrett

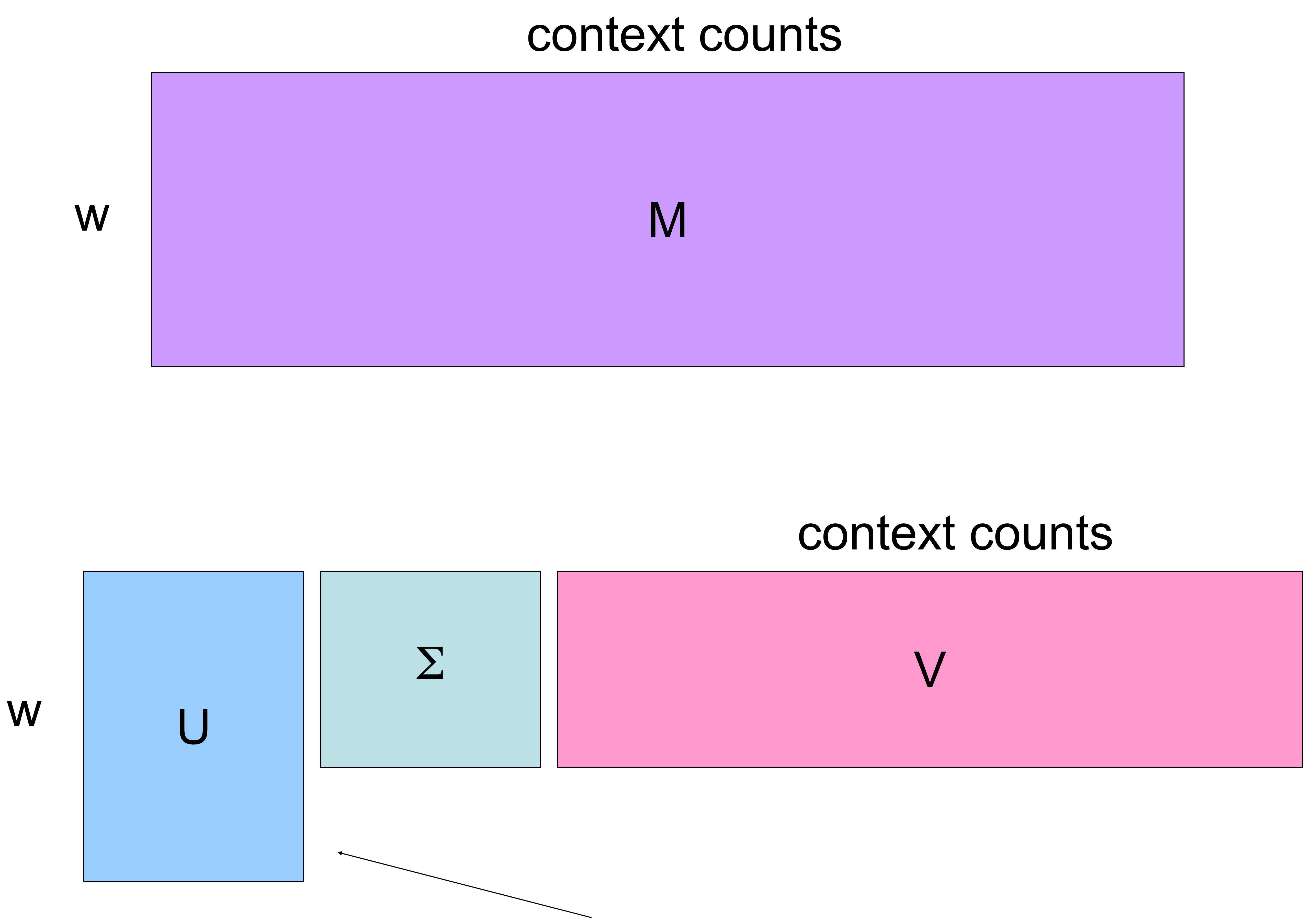
Distributional Similarity

- Key idea in clustering and embedding methods: characterize a word by the words it occurs with (cf Harris' distributional hypothesis, 1954)
  - "You can tell a word by the company it keeps." [Firth, 1957]
  - Harris / Chomsky divide in linguistic methodology



Vector Space Methods

- Treat words as points in R^n (eg Shuetze, 93)
  - Form matrix of co-occurrence counts
  - SVD or similar to reduce rank (cf LSA)
  - Cluster projections
  - People worried about things like: log of counts, U vs UΣ
- Today we'd call this an embedding method (it's basically GLoVe — Pennington et al. 2014), but we didn't want embeddings in 1993



Cluster these 50-200 dim vectors instead.

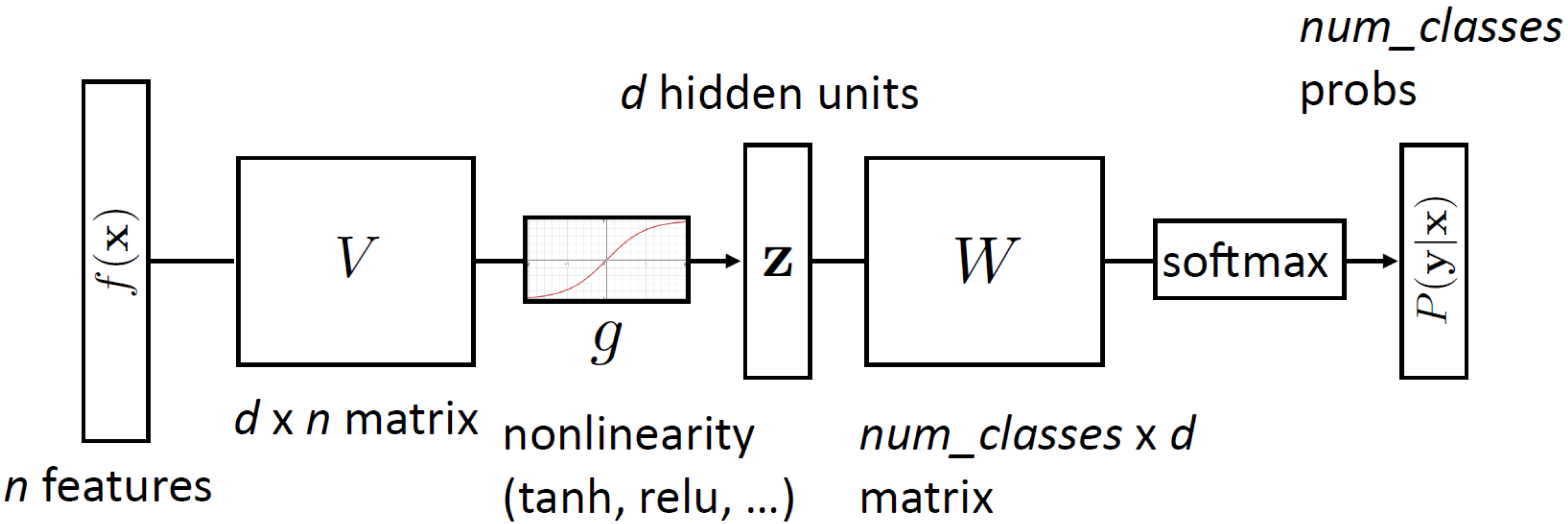
Neural Language Models

Neural LMs: Three Key Ideas

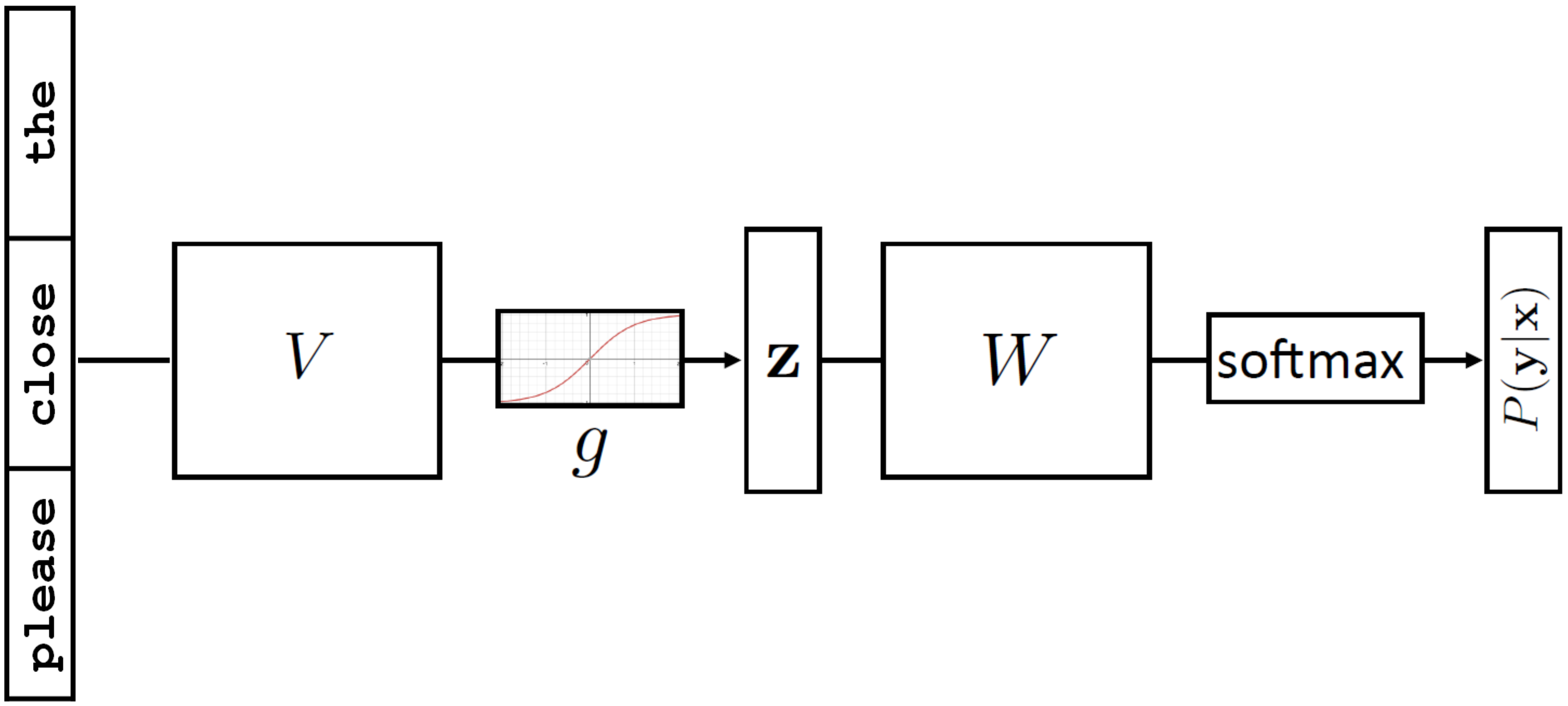
- Word embeddings
  - Different words are not entirely unrelated events
  - Words can be more and less similar, in complex ways
- Partially factored representations
  - Multiple semi-independent processes happen in parallel in language
  - It’s too expensive to track language in an unfactored way, and too inaccurate to assume everything of interest is independent
- Long distance dependencies
  - Information can be relevant without being local
  - Different notions of locality are important at different times

Reminder: Feedforward Neural Nets

$$P(\mathbf{y}|\mathbf{x}) = \text{softmax}(Wg(Vf(\mathbf{x})))$$

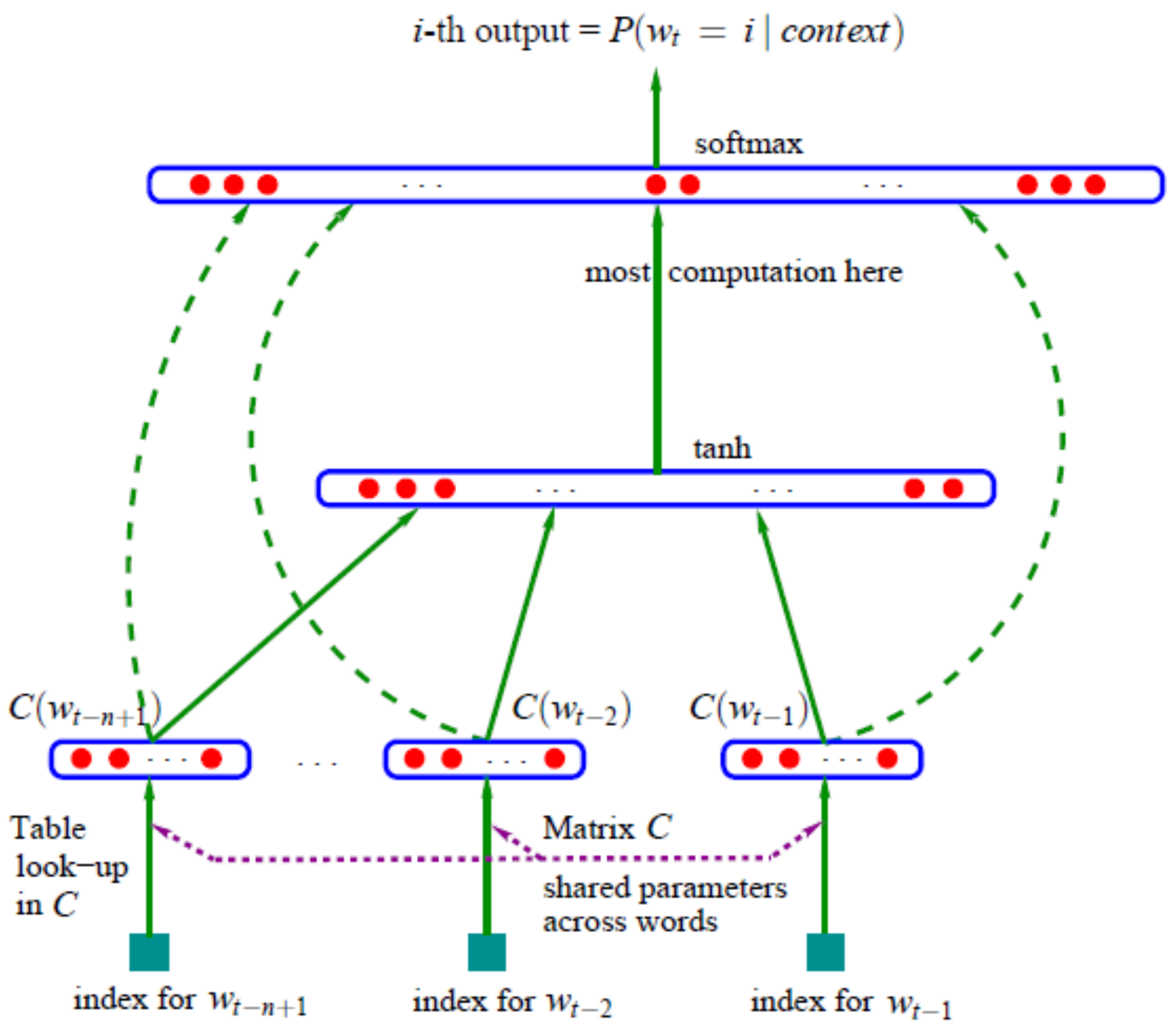


A Feedforward N-Gram Model?



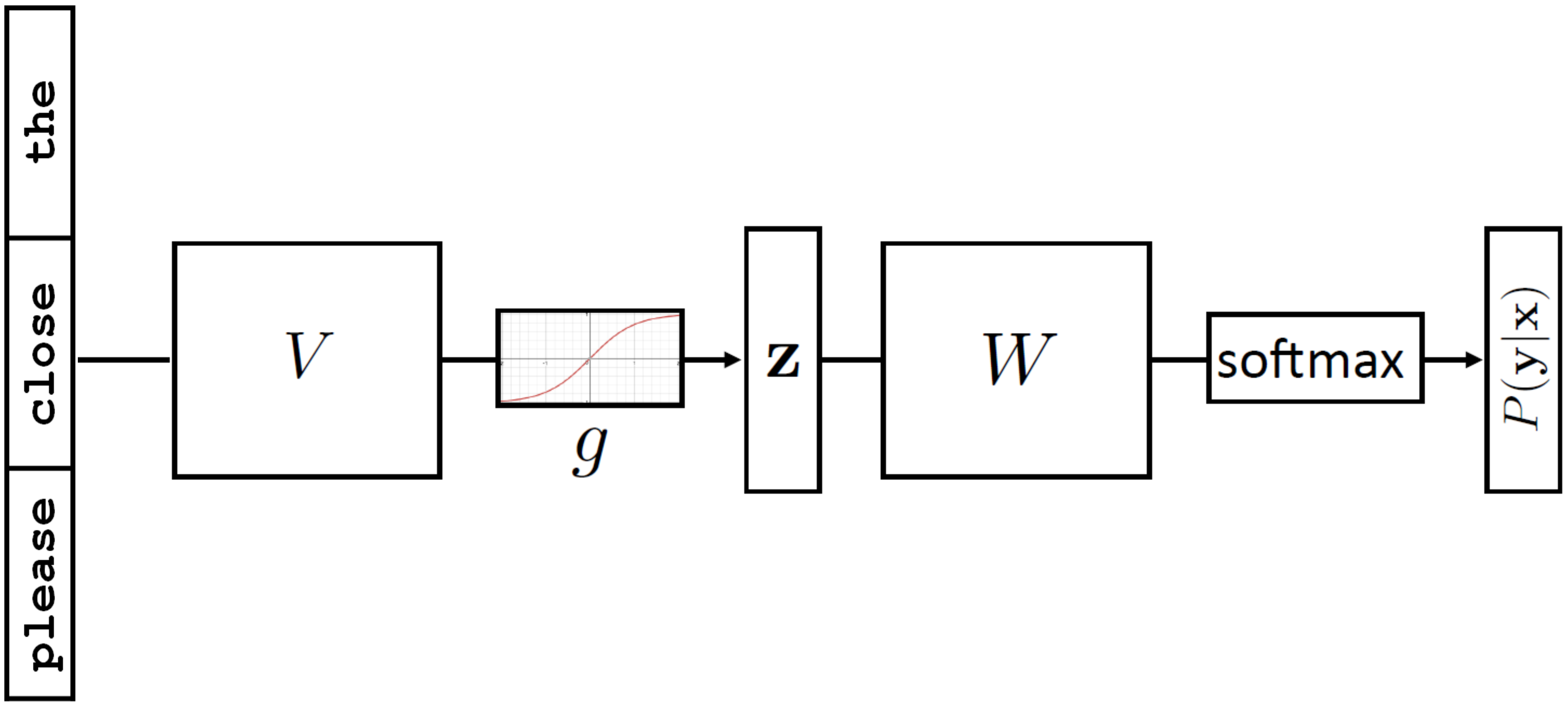
Early Neural Language Models

- Fixed-order feed-forward neural LMs
  - Eg Bengio et al 03
  - Allow generalization across contexts in more nuanced ways than prefixing
  - Allow different kinds of pooling in different contexts
  - Much more expensive to train



Bengio et al 03

Using Word Embeddings?



Using Word Embeddings

- Aproach 1: learn embeddings as parameters from data
  - Often works pretty well
- Approach 2: initialize (e.g. using GloVe), keep fixed
  - Fast because no need to learn or update parameters
- Approach 3: initialize (e.g. using GloVe), fine-tune
  - Works well for some tasks
- Modern approach: learning context embeddings
  - Will discuss later

Limitations of Fixed-Window NN LMs?

- What have we gained over N-Gram LMs?
- What have we lost?
- What have we not changed?