CSCI 5541: Natural Language Processing

Lecture 6: Langage Models: N-grams, Neural LM

Dongyeop Kang (DK), University of Minnesota

dongyeop@umn.edu | twitter.com/dongyeopkang | dykang.github.io

computer science & engineering

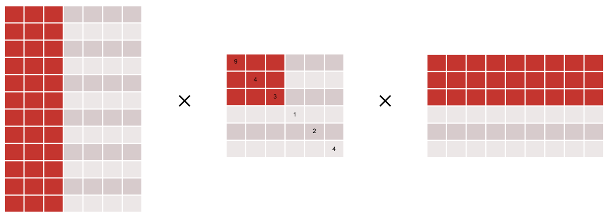
MINNESOTA NLP EST. 2021

University of Minnesota
Driven to Discover®

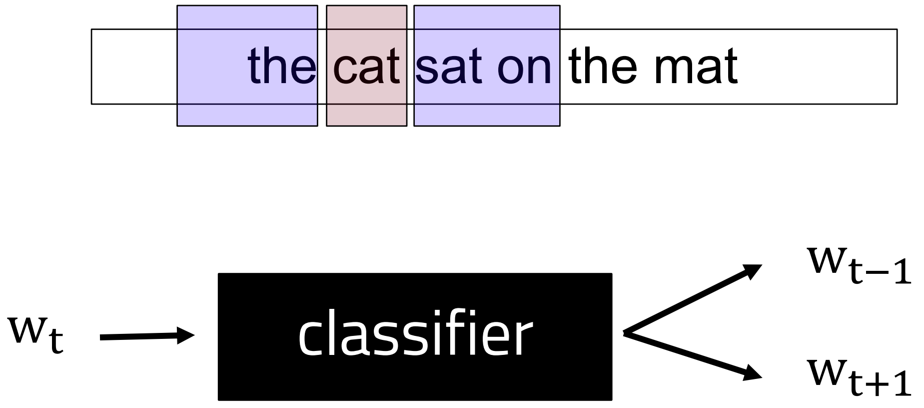
Count-based vs Prediction-based Methods

LSA, **HAL** (Lund & Burgess)
Hellinger-PCA (Rohde et al, Lebret & Collobert)

	Hamlet	Macbeth
knife	1	1
dog		
sword	2	2
love	64	
like	75	38

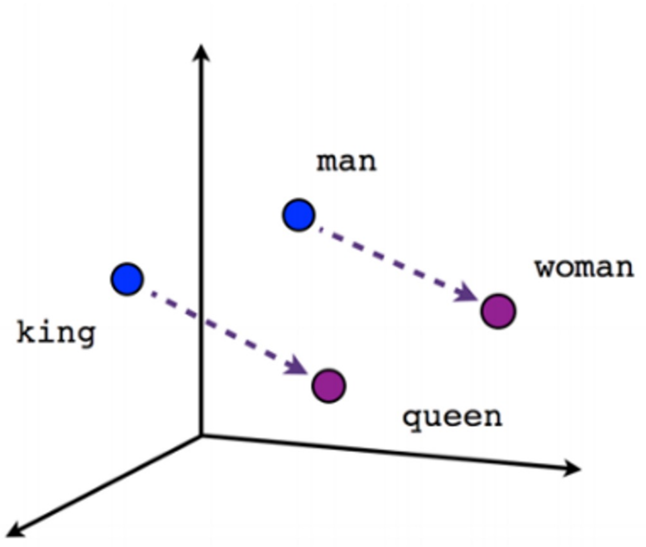


Skip-gram/CBOW (Mikolovet al)
NLM, HLBL, RNN (Bengioet al; Collobert & Weston; Huang et al; Mnih & Hinton)

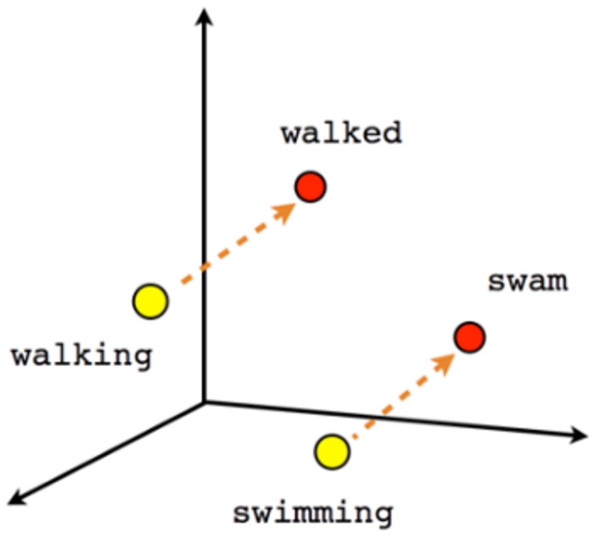


Evaluations

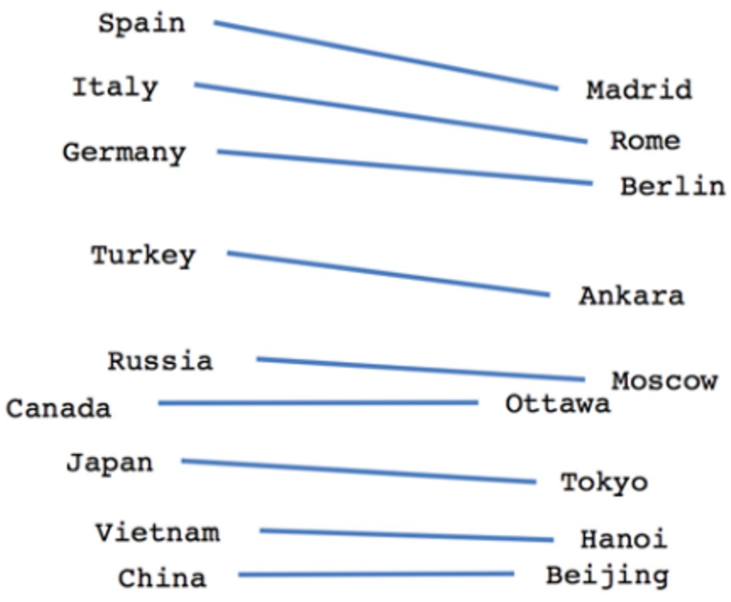
Recap



Male-Female



Verb tense



Country-Capital

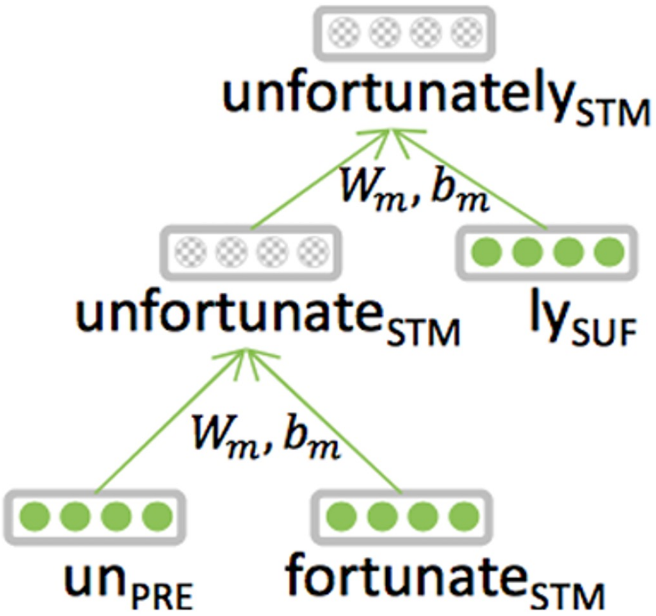
Limitations of Embeddings

Recap

- Sensitive to **superficial differences** (dog / dogs)
  - E.g. misspellings: “minuscule” → “miniscule”
  - E.g. compounded/prefixed/suffixed words split into “wrong” subwords “descheduled” ⇒ [“des”, “##ched”, “##uled”]
- **Not necessarily coordinated** with knowledge or across languages
- Can encode **bias** (encode stereotypical gender roles, racial biases)

Limitations and Solutions

Recap



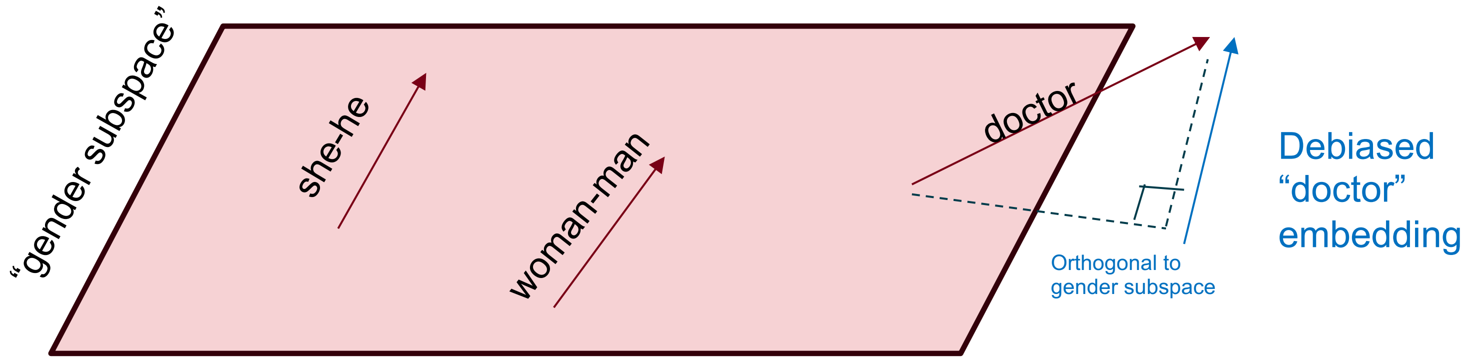
Morpheme-based (Luong et al. 2013)

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

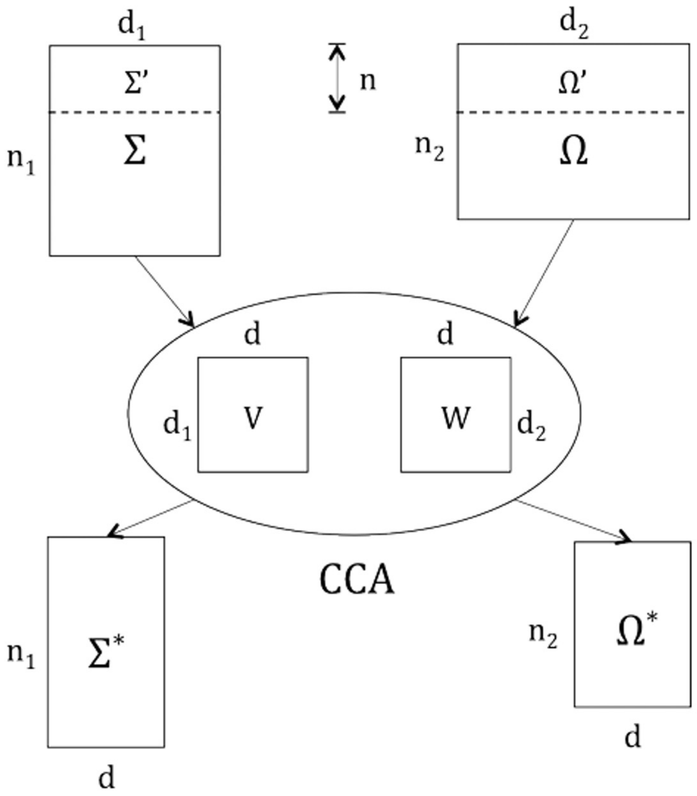
Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

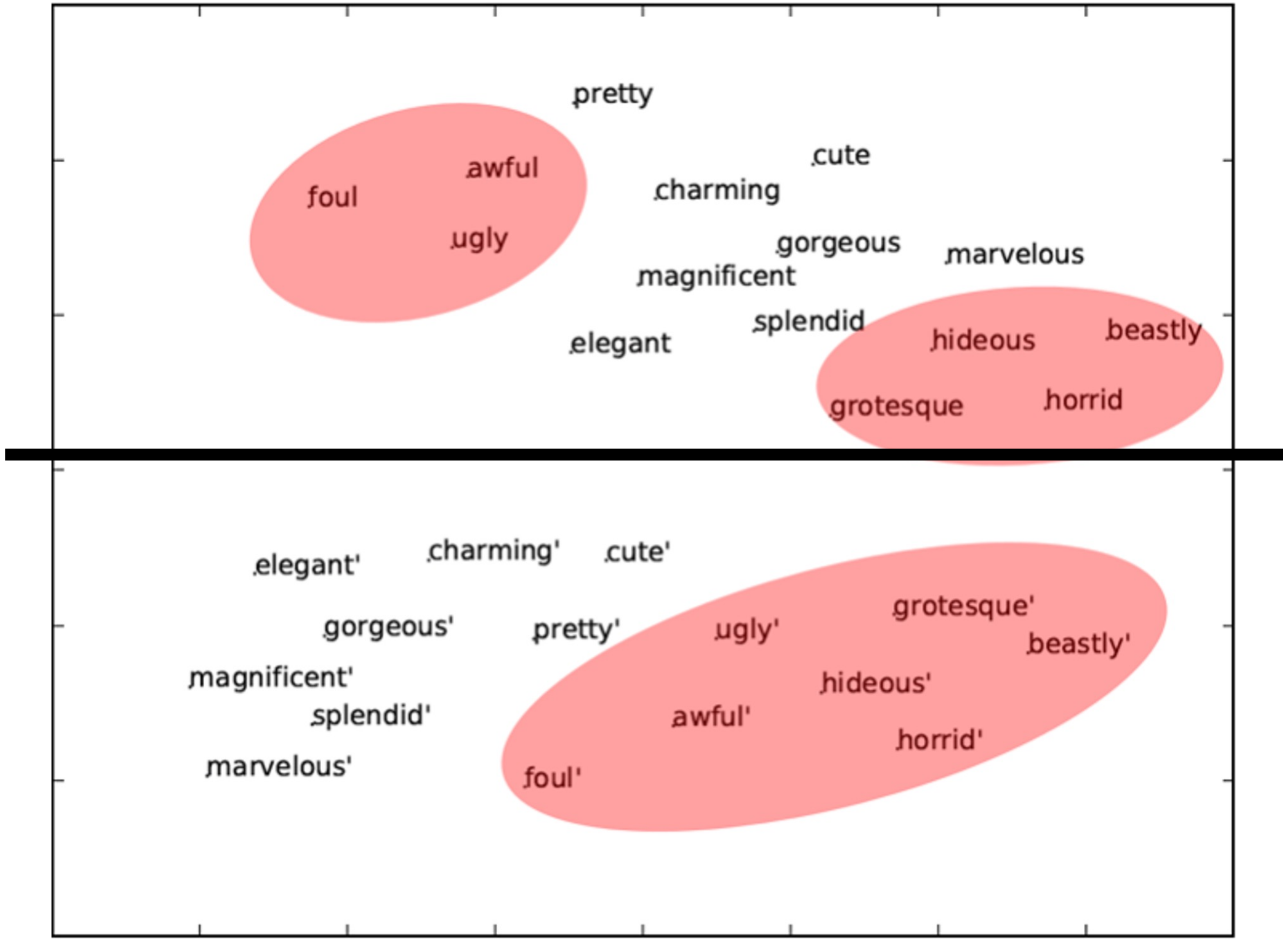


[Bolukbasi et al. 2016]

Multilingual Coordination of Embeddings using dictionaries



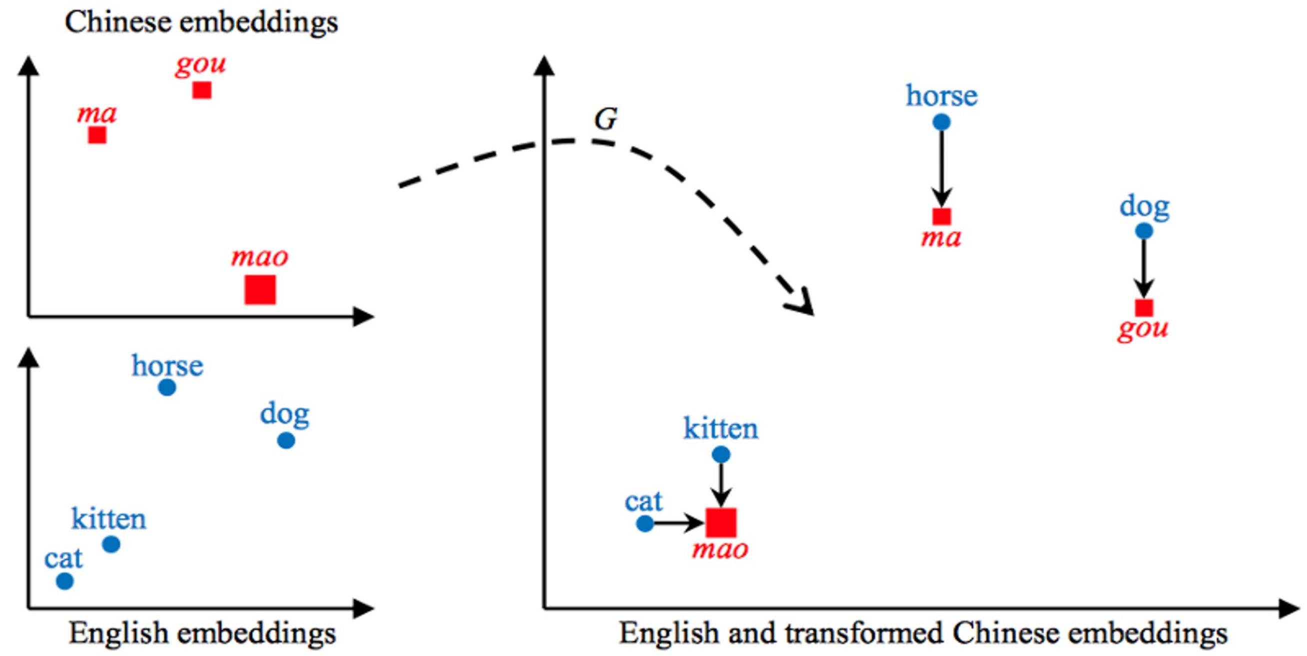
Improving Vector Space Word Representations Using Multilingual Correlation (Faruqui & Dyer, 2014)



Monolingual (top) and multilingual (bottom) word projections of the antonyms (shown in red) and synonyms of “beautiful”

Unsupervised Coordination of Embeddings

- In some cases, we can do it with no dictionary at all!
  - Just use identical words, e.g. the digits (Artexte et al. 2017)
  - Or, just match distributions (Zhang et al. 2017)

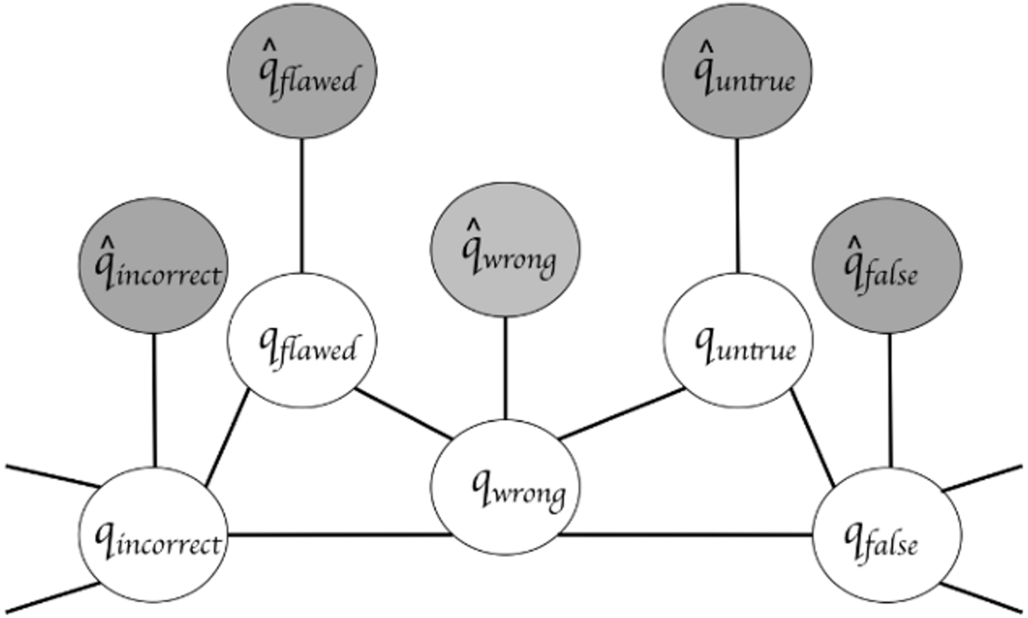


Retrofitting of Embeddings to Existing Lexicons

- Make word vectors to match with existing lexicon like WordNet (Faruqui et al. 2015)

Word Embeddings

WordNet



$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \| q_i - \hat{q}_i \|^2 + \sum_{(i,j) \in E} \beta_{ij} \| q_i - q_j \|^2 \right]$$

semantic to model temporal word analogy or relatedness (Szymanski, 2017; Rosin et al., 2017) or to capture the dynamics of semantic relations (Kutuzov et al., 2017)

Different kinds of encoding “context”

Recap

- ~~Count-based~~
  - PMI, TF-IDF
- ~~Distributed prediction-based (type) embeddings~~
  - Word2vec, GloVe, Fasttext
- Distributed contextual (token) embeddings from **language models**
  - ELMo, BERT, GPT
- ~~Many more variants~~
  - Multilingual / multi-sense / syntactic embeddings, etc

Outline

- Language modeling
- Applications of language models
- How to estimate $P(w)$ from data? Ngram Language Model (LM)
- Advanced techniques for ngram LM
- Ngram LM vs Neural LM

Which sentence is more natural?

"DK me Call"

"me Call DK"

"Call me DK"

Language modeling

- Provide a way to quantity the likelihood of a sequence
  - i.e., plausible sentences
- Vocabulary (V) is a finite set of discrete symbols (e.g., words, characters);
  - ~170K words for English, ~150K words for Russian, ~1.1M words for Korean, ~85K words for Chinese
- V^+ is the infinite set of sequences of symbols from V; each sequence ends with STOP
  - A sentence of k words: $V \ * V \, ..* \, V = V^k$ e.g., $170{,}000^{100}$ for English 100-length sentence

sequence

$$P(w) = P\ (w_1, \ldots w_n)$$

$$P(\text{"Call me DK"})$$
$$= P(w_1 = \text{"Call"}, w_2 = \text{"me"}, w_2 = \text{"DK"}) \times P(\text{"STOP"})$$

$$\sum_{w \in V^+} P(w) = 1 \qquad 0 \le P(w) \le 1$$

over all the possible sequences of words

Which sentence is more natural?

"Call me DK"

"DK me Call"

$P(\text{"Call me DK"}) = 10^{-5}$

$P(\text{"DK me call"}) = 10^{-15}$

Use Cases of Language Model

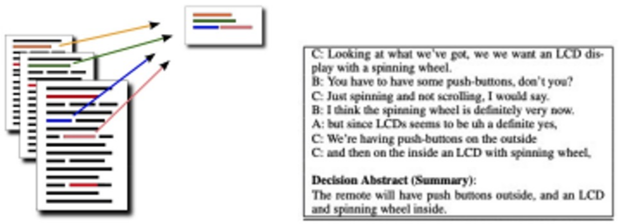
- Provide a way to quantity the likelihood of a sequence i.e., plausible sentences
  - Probability distributions over sentences (i.e., word sequences)

    $P(w) = P\ (w_1, \dots w_n)$
- Can use them to generate strings
  - $P(w_k \mid w_2 w_3 w_4 \dots w_{k-1})$
- Rank possible sentences
  - $P(\text{"Today is Thursday'}) > P(\text{"Thursday Today is '})$
  - $P(\text{"Today is Thursday'}) > P(\text{"Today is Minneapolis'})$

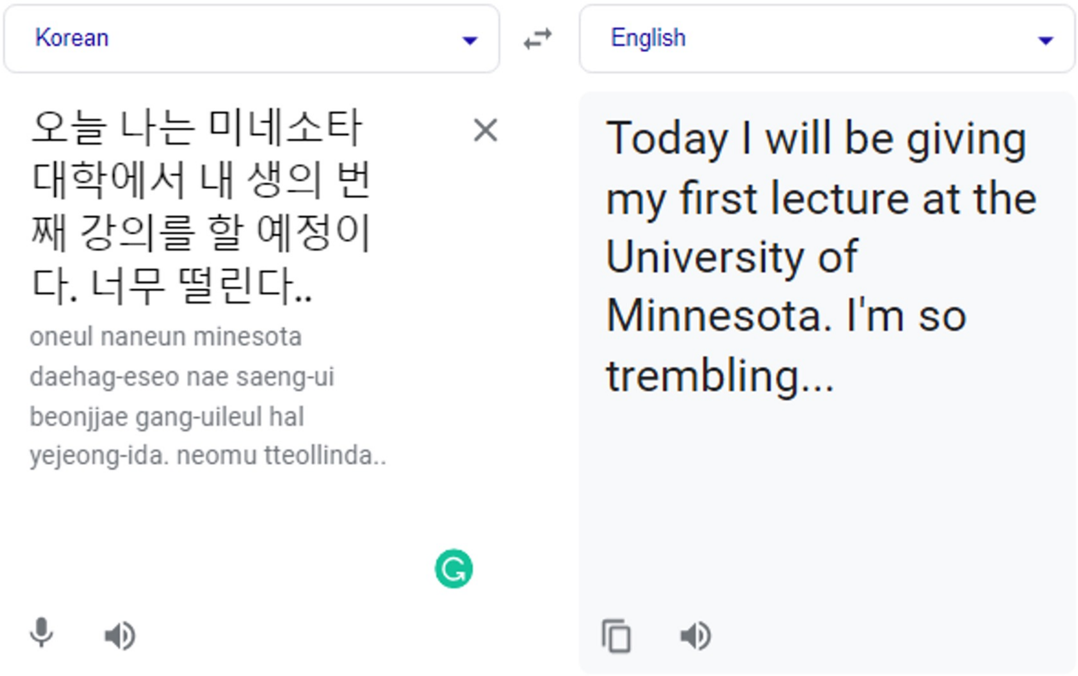
Applications of language models

What is natural language generation?

- NLP = Natural Language Understanding (NLU) + Natural Language Generation (NLG)
- NLG focuses on systems that produce coherent and useful language output for human consumption
- Deep Learning is powering (some) next-gen NLG systems



Machine Translation



Fluency of the translation

$$P(Y \mid X) + a * P(Y)$$

Optical Character Recognition (OCR)

To ſee great *Pompey* paſſe the ſtreets of Rome:
And when you ſaw his Chariot but appeare,
Haue you not made an Vniuerſall ſhout,
That Tyber trembled vnderneath her bankes
To heare the replication of your ſounds,
Made in her Concaue Shores?

to fee great Pompey paffe the Areets of Rome:

to **s**ee great Pompey pa**ss**e the **st**reets of Rome:

Speech Recognition

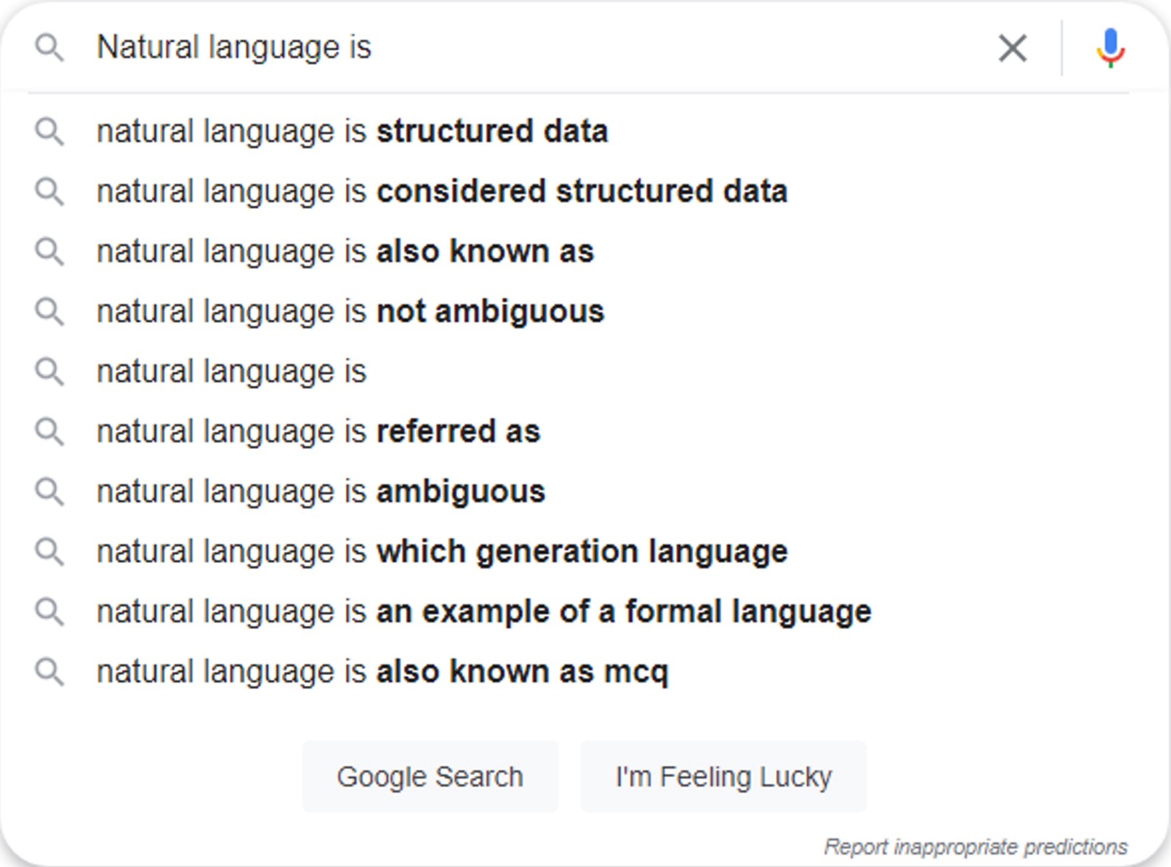
'Scuse me while I kiss this guy

'Scuse me while I kiss the sky ✓

'Scuse me while I kiss this fly

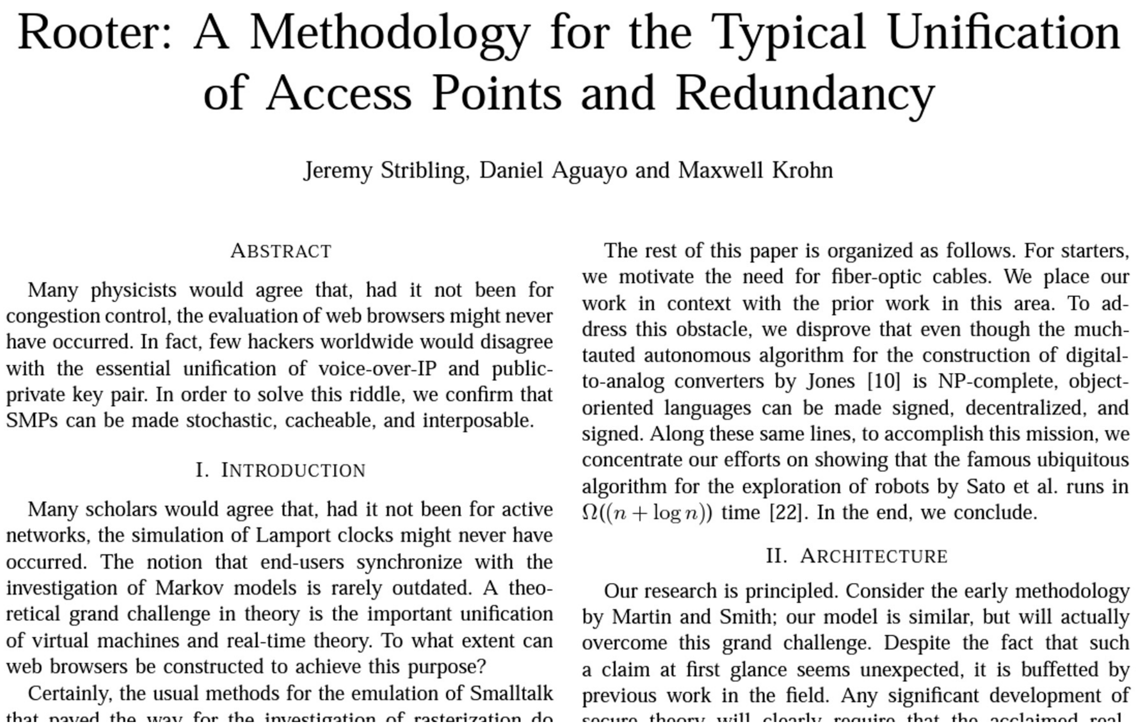
'Scuse me while my biscuits fry

Automatic Completion



$$P(w_k \mid w_2 w_3 w_4 \dots w_{k-1})$$

Language Generation



Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn

ABSTRACT

Many physicists would agree that, had it not been for congestion control, the evaluation of web browsers might never have occurred. In fact, few hackers worldwide would disagree with the essential unification of voice-over-IP and public-private key pair. In order to solve this riddle, we confirm that SMPs can be made stochastic, cacheable, and interposable.

I. INTRODUCTION

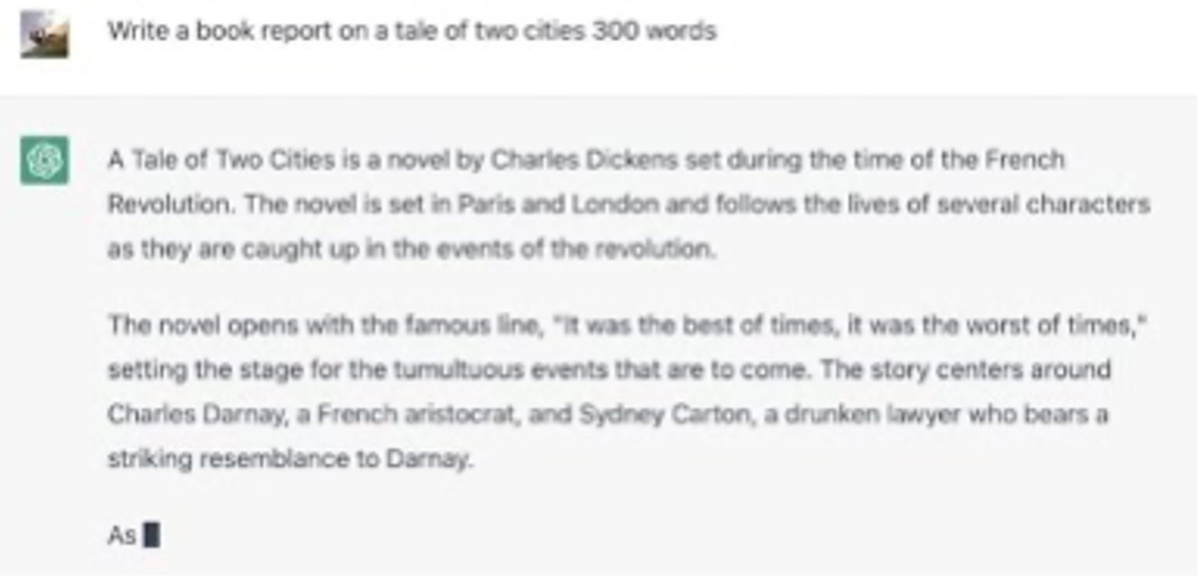
Many scholars would agree that, had it not been for active networks, the simulation of Lamport clocks might never have occurred. The notion that end-users synchronize with the investigation of Markov models is rarely outdated. A theoretical grand challenge in theory is the important unification of virtual machines and real-time theory. To what extent can web browsers be constructed to achieve this purpose?

Certainly, the usual methods for the emulation of Smalltalk that paved the way for the investigation of rasterization do

The rest of this paper is organized as follows. For starters, we motivate the need for fiber-optic cables. We place our work in context with the prior work in this area. To address this obstacle, we disprove that even though the much-tauted autonomous algorithm for the construction of digital-to-analog converters by Jones [10] is NP-complete, object-oriented languages can be made signed, decentralized, and signed. Along these same lines, to accomplish this mission, we concentrate our efforts on showing that the famous ubiquitous algorithm for the exploration of robots by Sato et al. runs in $\Omega((n + \log n))$ time [22]. In the end, we conclude.

II. ARCHITECTURE

Our research is principled. Consider the early methodology by Martin and Smith; our model is similar, but will actually overcome this grand challenge. Despite the fact that such a claim at first glance seems unexpected, it is buffetted by previous work in the field. Any significant development of secure theory will clearly require that the acclaimed real-



https://pdos.csail.mit.edu/archive/scigen/

Dialogue Generation

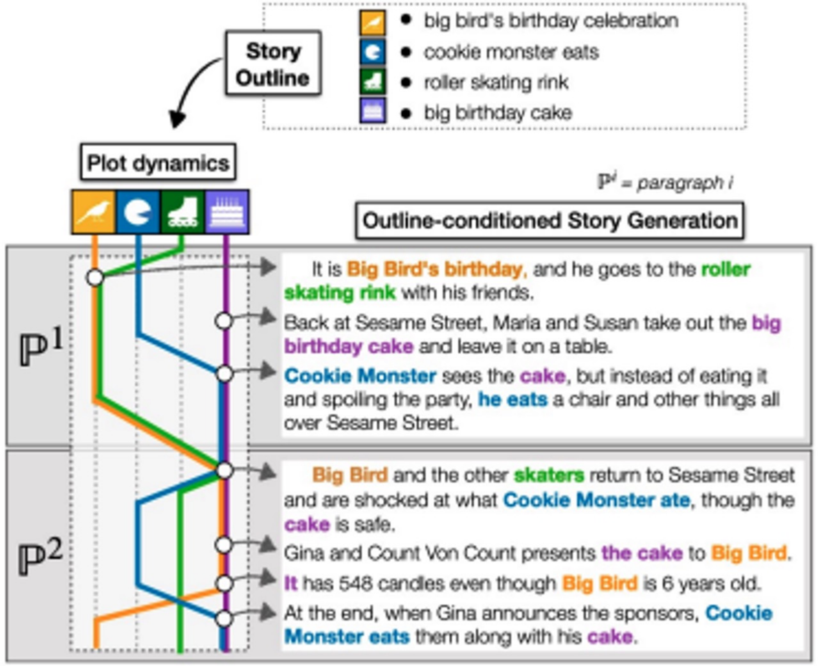
Q: How many rainbows does it take to jump from Hawaii to seventeen?
A: It takes two rainbows to jump from Hawaii to seventeen.

Q: Which colorless green ideas sleep furiously?
A: Ideas that are colorless, green, and sleep furiously are the ideas of a sleep furiously.

Q: Do you understand these questions?
A: I understand these questions.

Giving GPT-3 a Turing Test, Kevin Lacker's blog, https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

More interesting NLG uses



Creative story generation

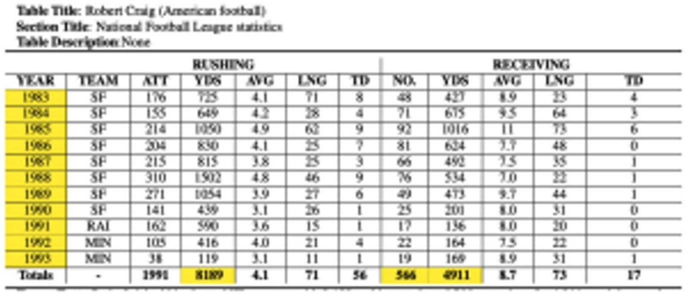


Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description: None

		RUSHING					RECEIVING				
YEAR	TEAM	ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Data/Table to text

Two children are sitting at a table in a restaurant. The children are one little girl and one little boy. The little girl is eating a pink frosted donut with white icing lines on top of it. The girl has blonde hair and is wearing a green jacket with a black long sleeve shirt underneath. The little boy is wearing a black zip up jacket and is holding his finger to his lip but is not eating. A metal napkin dispenser is in between them at the table. The wall next to them is white brick. Two adults are on the other side of the short white brick wall. The room has white circular lights on the ceiling and a large window in the front of the restaurant. It is daylight outside.

Visual description


ST
Can you write out an Adobe After Effects expression to make a shape layer wiggle when a null object is within 50 pixels of the shape's anchor point.

Language modeling is the task of estimating $P(w)$

How to estimate $P(w)$ from data?

Chain rule (of probability)

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5) &= P(x_1) \\ &\times P(x_2|x_1) \\ &\times P(x_3|x_1, x_2) \\ &\times P(x_4|x_1, x_2, x_3) \\ &\times P(x_5|x_1, x_2, x_3, x_4) \end{aligned}$$

"The mouse that the cat that the dog that the man frightened and chased ran away."

"The mouse that the cat that the dog that the man frightened and chased ran away."

Easy

$P(\text{"The"})$ $\quad$ $P(x_1)$

$P(\text{"mouse"} \mid \text{"The"})$ $\quad$ $P(x_2 \mid x_1)$

$P(\text{"that"} \mid \text{"The"}, \text{"mouse"})$ $\quad$ $P(x_3 \mid x_1, x_2)$

$P(\text{"the"} \mid \text{"The"}, \text{"mouse"}, \text{"that"})$ $\quad$ $P(x_4 \mid x_1, x_2, x_3)$

$P(\text{"away"} \mid \text{"The"}, \text{"mouse"}, \text{"that"}, \text{"the"}, \text{"cat"} \ \ldots)$ $\quad$ $P(x_n \mid x_1, x_2 \ldots x_{n-1})$

Hard

Markov assumption

$$
\begin{aligned}
&= \cancel{P(x_1)} \\
&\times P(x_2|x_1) \\
&\times P(x_3|\cancel{x_1,}\, x_2) \\
&\times P(x_4|\cancel{x_1,}\,\cancel{x_2,}\, x_3) \\
&\times P(x_5|\cancel{x_1,}\,\cancel{x_2,}\,\cancel{x_3,}\, x_4)
\end{aligned}
\qquad
\begin{aligned}
&= \cancel{P(x_1)} \\
&\times \cancel{P(x_2|x_1)} \\
&\times P(x_3|x_1, x_2) \\
&\times P(x_4|\cancel{x_1,}\, x_2, x_3) \\
&\times P(x_5|\cancel{x_1,}\,\cancel{x_2,}\, x_3, x_4)
\end{aligned}
$$

first-order $\quad P(x_i|\, x_1, x_2 \ldots x_{i-1}\,) \approx P(x_i|\, x_{i-1}\,)$

second-order $\quad P(x_i|\, x_1, x_2 \ldots x_{i-1}\,) \approx P(x_i|\, x_{i-2}, x_{i-1}\,)$

Markov assumption

Bi-gram model
(first-order markov)

$$P(w) = \prod_{i=1}^{n} P(w_i | w_{i-1}) \times P(\text{STOP} \mid w_n)$$

Tri-gram model
(second-order markov)

$$P(w) = \prod_{i=1}^{n} P(w_i | w_{i-2}, w_{i-1}) \times P(\text{STOP} \mid w_{n-1}, w_n)$$

Bi-gram model
(first-order markov)

$$P(\text{"}The\text{"} \mid \text{START}_1, \text{START}_2)$$

$$P(\text{"}mouse\text{"} \mid \text{START}_2, \text{"}The\text{"})$$

$$P(\text{"}that\text{"} \mid \text{"}The\text{"}, \text{"}mouse\text{"})$$

$$P(\text{"}the\text{"} \mid \text{"}mouse\text{"}, \text{"}that\text{"})$$

...

$$P(\text{"}away\text{"} \mid \text{"}chased\text{"}, \text{"}ran\text{"})$$

$$P(\text{STOP} \mid \text{"}ran\text{"}, \text{"}away\text{"})$$

"The mouse that the cat that the dog that the man frightened and chased ran away."

Estimation from data

Uni-gram

$$\prod_{i=1}^{n} P(w_i) \times P(STOP)$$

$$\frac{c(w_i)}{N}$$

Bi-gram

$$\prod_{i=1}^{n} P(w_i | w_{i-1}) \times P(STOP \mid w_n)$$

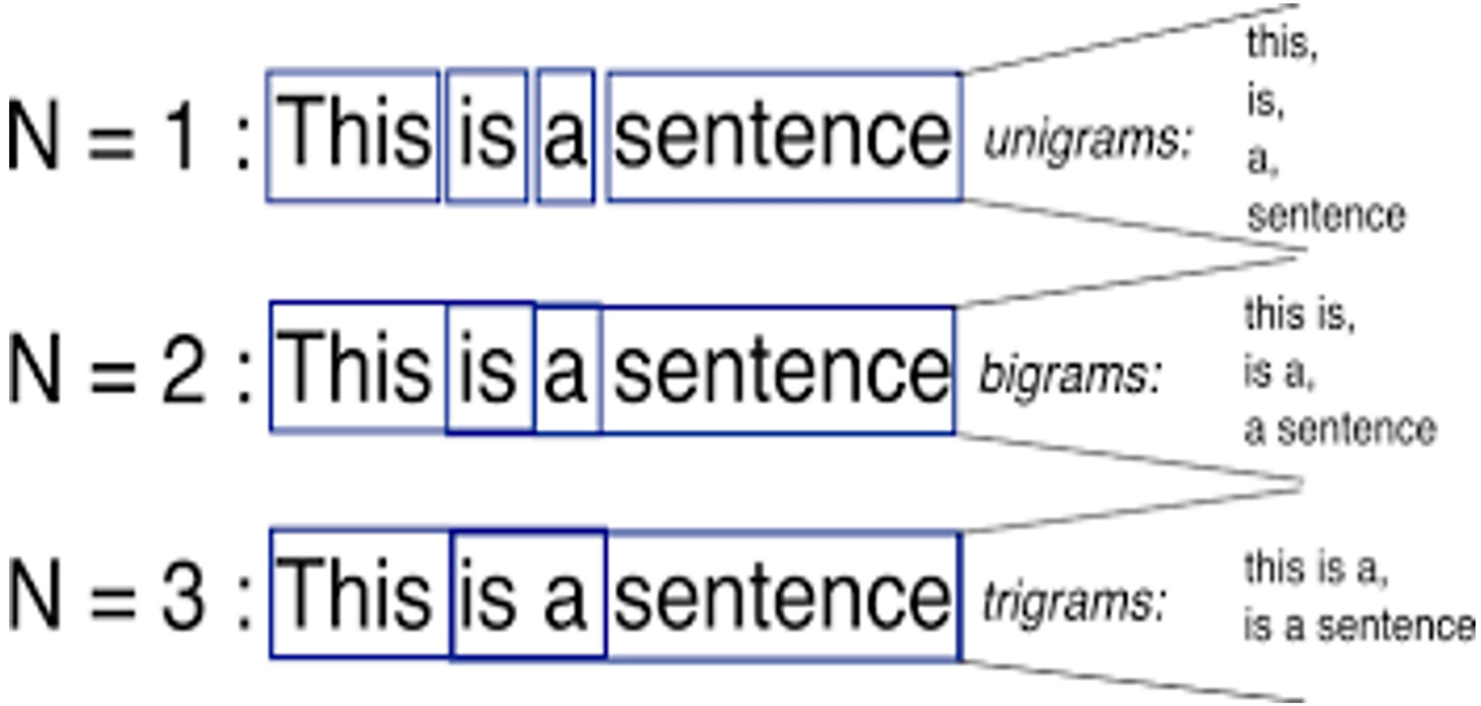
$$\frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Tri-gram

$$\prod_{i=1}^{n} P(w_i | w_{i-2}, w_{i-1}) \times P(STOP \mid w_{n-1} w_n)$$

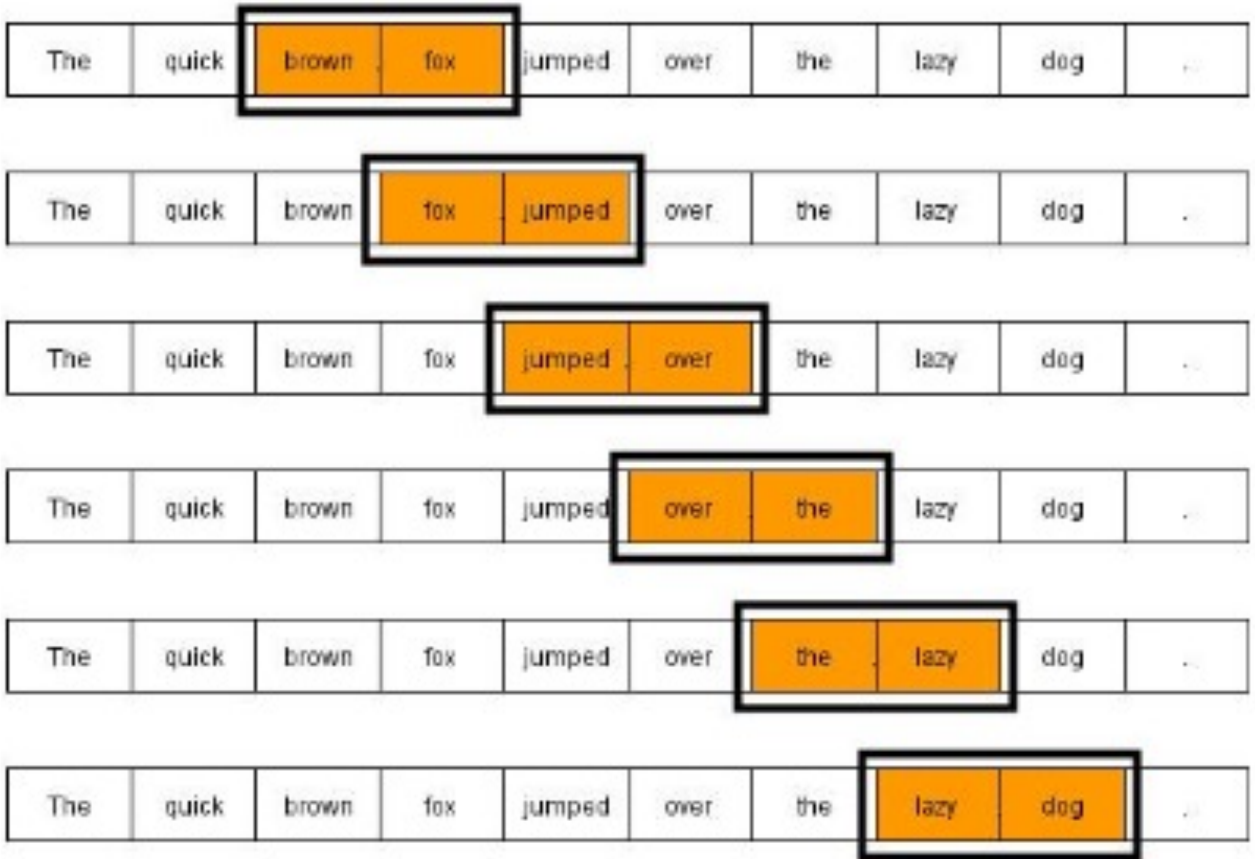
$$\frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

Estimation from data



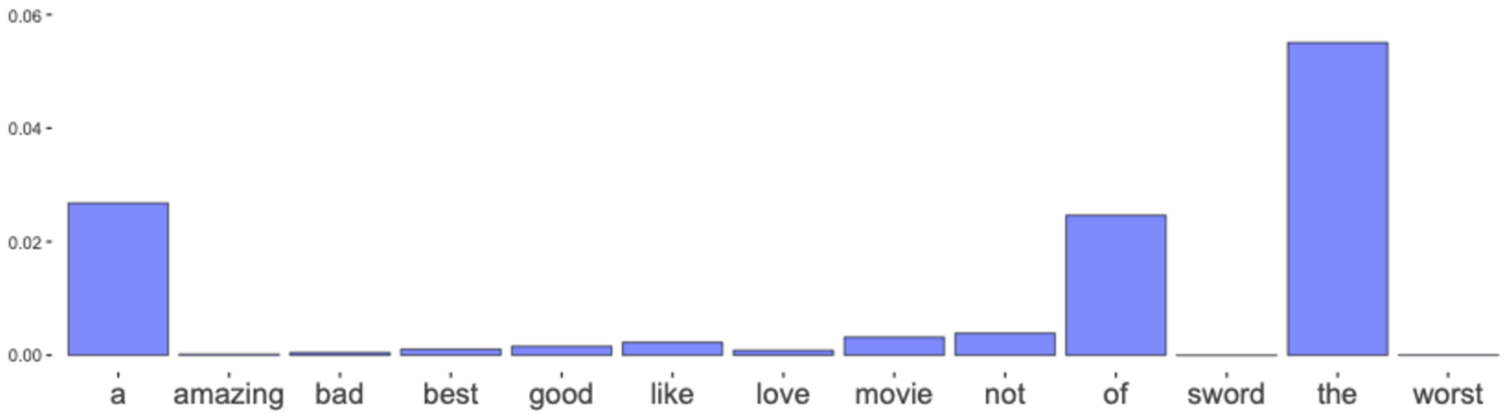
Estimation from data

$$c(w_{i-1}, w_i)$$



Generating from language model

- What we learn in estimating language models is P (word | context), where context is the previous n-1 words (for ngram of order n)
- We have one multinomial over the vocabulary including STOP for each context



$$P(\text{“the”} \mid \text{“mouse”}, \text{”that”})$$

Part of A Unigram Distribution trained on academic papers

[rank 1]
p(the) = 0.038
p(of) = 0.023
p(and) = 0.021
p(to) = 0.017
p(is) = 0.013
p(a) = 0.012
p(in) = 0.012
p(for) = 0.009
…

…
[rank 1001]
p(joint) = 0.00014
p(relatively) = 0.00014
p(plot) = 0.00014
p(DEL1SUBSEQ) = 0.00014
p(rule) = 0.00014
p(62.0) = 0.00014
p(9.1) = 0.00014
p(evaluated) = 0.00014
…

Generated text from a uni-gram model

first, from less the This different 2004), out which goal 19.2 Model their It ~(i?1), given 0.62 these (x0; match 1 schedule. x 60 1998. under by Notice we of stated CFG 120 be 100 a location accuracy If models note 21.8 each 0 WP that the that Nov?ak. to function; to [0, to different values, model 65 cases. said - 24.94 sentences not that 2 In to clustering each K&M 100 Boldface X))] applied; In 104 S. grammar was (Section contrastive thesis, the machines table -5.66 trials: An the textual (family applications.Wehave for models 40.1 no 156 expected are neighborhood

Generated text from a bi-gram model

e. (A.33) (A.34) A.5 ModelS are also been completely surpassed in performance on drafts of online algorithms can achieve far more so while substantially improved using CE. 4.4.1 MLEasaCaseofCE 71 26.34 23.1 57.8 K&M 42.4 62.7 40.9 44 43 90.7 100.0 100.0 100.0 15.1 30.9 18.0 21.2 60.1 undirected evaluations directed DEL1 TRANS1 neighborhood. This continues, with supervised init., semisupervised MLE with the METU- SabanciTreebank 195 ADJA ADJD ADV APPR APPRART APPO APZR ART CARD FM ITJ KOUI KOUS KON KOKOM NN NN NN IN JJ NNTheir problem is y x. The evaluation offers the hypothesized link grammar with a Gaussian

Generated text from a tri-gram model

top(xI ,right,B). (A.39) vine0(X, I) rconstit0(I 1, I). (A.40) vine(n). (A.41) These equations were presented in both cases; these scores u<AC>into a probability distribution is even smaller(r =0.05). This is exactly fEM. During DA, is gradually relaxed. This approach could be efficiently used in previous chapters) before training (test) K&MZeroLocalrandom models Figure4.12: Directed accuracy on all six languages. Importantly, these papers achieved state- of-the-art results on their tasks and unlabeled data and the verbs are allowed (for instance) to select the cardinality of discrete structures, like matchings on weighted graphs (McDonald et al., 1993) (35 tag types, 3.39 bits). The Bulgarian,

Evaluation for Language Models

- The best evaluation metrics are external
  - How does a better language model influence the application you care about?
  - E.g.,

    machine translation (BLEU score)

    sentiment classification (F1 score)

    speech recognition (word error rate)

(Intrinsic) Evaluation

- A good language model should judge unseen real language to have high probability
- Perplexity = inverse probability of test data, averaged by word
    - Better models have lower perplexity
- To be reliable, the test data must be truly unseen (including knowledge of its vocabulary)

$$\text{Perplexity} = \sqrt[N]{\frac{1}{P(w_1, \ldots, w_n)}}$$

$$\sqrt[N]{\frac{1}{\prod_i^N P(w_i)}} = \left(\prod_i^N P(w_i)\right)^{-\frac{1}{N}}$$

$$\sqrt[N]{\frac{1}{\prod_i^N P(w_i)}} = \left(\prod_i^N P(w_i)\right)^{-\frac{1}{N}}$$

$$= \exp log \left(\prod_i^N P(w_i)\right)^{-\frac{1}{N}}$$

$$= \exp\left(-\frac{1}{N} \log \prod_i^N P(w_i)\right)$$

Perplexity $= \exp\left(-\frac{1}{N} \sum_i^N \log P(w_i)\right)$

$$\sqrt[N]{\frac{1}{\prod_i^N P(w_i)}} = \left(\prod_i^N P(w_i)\right)^{-\frac{1}{N}}$$

$$= \exp log \left(\prod_i^N P(w_i)\right)^{-\frac{1}{N}}$$

$$= \exp\left(-\frac{1}{N}\log\prod_i^N P(w_i)\right)$$

Perplexity $= \exp\left(-\frac{1}{N}\sum_i^N \boxed{\log P(w_i)}\right)$

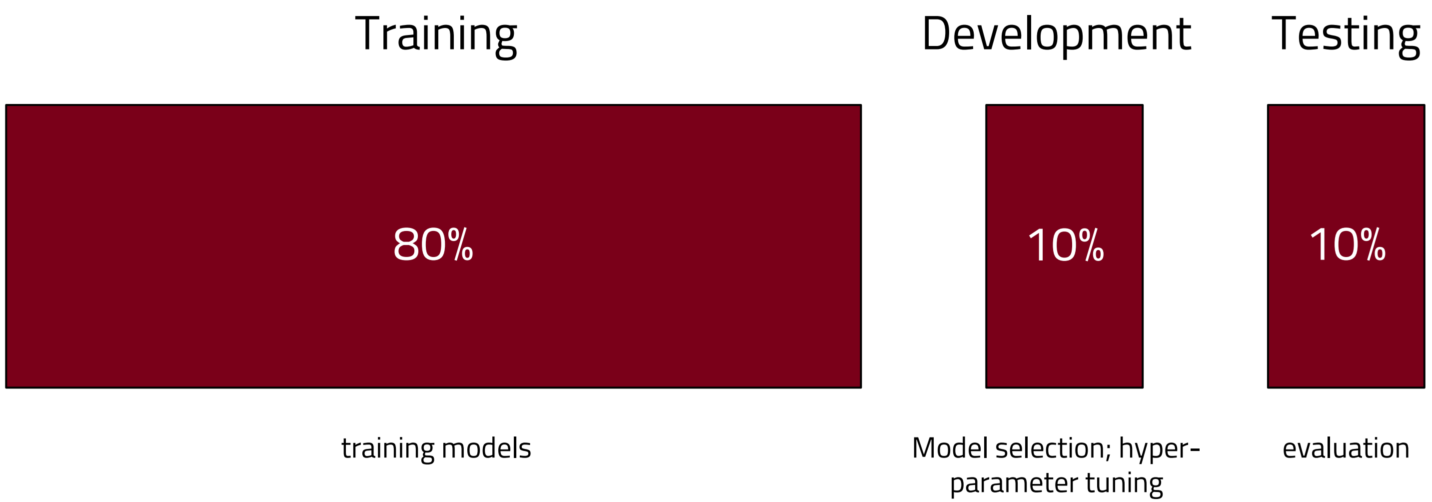
Bi-gram

$\boxed{P(w_i \mid w_{i-1})}$

Tri-gram

$\boxed{P(w_i \mid w_{i-2}, w_{i-1})}$

Intrinsic Evaluation



Perplexity

Model	Unigram	Bigram	Trigram
Perplexity	962	170	109

On PennTreeBank test set

Advanced techniques for ngram LM

Data sparsity

- Training data is a small (and biased) sample of the creativity of language.

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

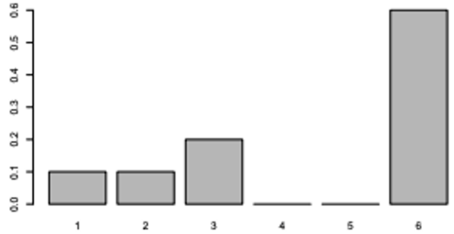
$$\frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

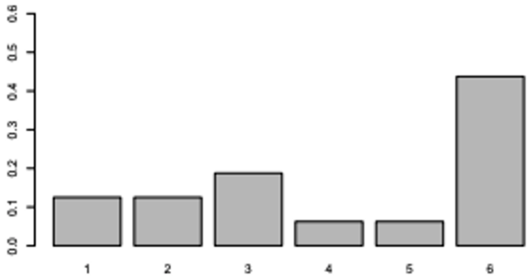
SLP3 4.1

Additive Smoothing

Uni-gram

$$\frac{c(w_i)}{N} \Rightarrow \frac{c(w_i)+\alpha}{N+V\alpha}$$





smoothing with α =1

Bi-gram

$$\frac{c(w_{i-1},w_i)}{c(w_{i-1})} \Rightarrow \frac{c(w_{i-1},w_i)+\alpha}{c(w_{i-1})+V\alpha}$$

Kneser-ney smoothing

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University, 1998.

Interpolation over different LMs

- As ngram order rises, we have the potential for higher precision but also higher variability in our estimates.

- A linear interpolation of any two language models p and q (with $\lambda \in [0,1]$) is also a valid language model

q = LM of political speeches

$$\lambda p + (1 - \lambda) q$$

p = LM of web

Interpolation over higher-order LMs

- How do we pick the best values of λ?
  - Grid search over Dev set

$$
\begin{aligned}
P(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1 P(w_i \mid w_{i-2}, w_{i-1}) \\
+ \lambda_2 P(w_i \mid w_{i-1}) \\
+ \lambda_3 P(w_i)
\end{aligned}
$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

Stupid backoff

back off to lower order ngram if the higher order is not observed.

if full sequence observed

$$S(w_i \mid w_{i-k+1}, \ldots, w_{i-1}) = \frac{c(w_{i-k+1}, \ldots, w_i)}{c(w_{i-k+1}, \ldots, w_{i-1})}$$

Otherwise

$$= \lambda S(w_i \mid w_{i-k+2}, \ldots, w_{i-1})$$

Cheap to calculate; works well when there is a lot of data

Brants et al. (2007), "Large Language Models in Machine Translation"

HW2: Authorship attribution using ngram language models (LMs)

- In your HW2
  - Smoothing and Backoff for handling sparsity
  - Interpolation between two ngram language models
  - Evaluating perplexity on held-out data
  - Generating a sentence from a trained model
  - Compare generative classifier (LMs) and discriminative classifier for authorship attribution
- Prerequisite:
  - Carefully read Section 3.5 of Jurafsky and Martin
  - Get used to NLTK's LM package
  - Extend your binary Huggingface-based classifier to multi-class

Ngram LM vs Neural LM

Neural LM

$$x = [v(w_1); \dots v(w_k)]$$

Concatenation (k x V)

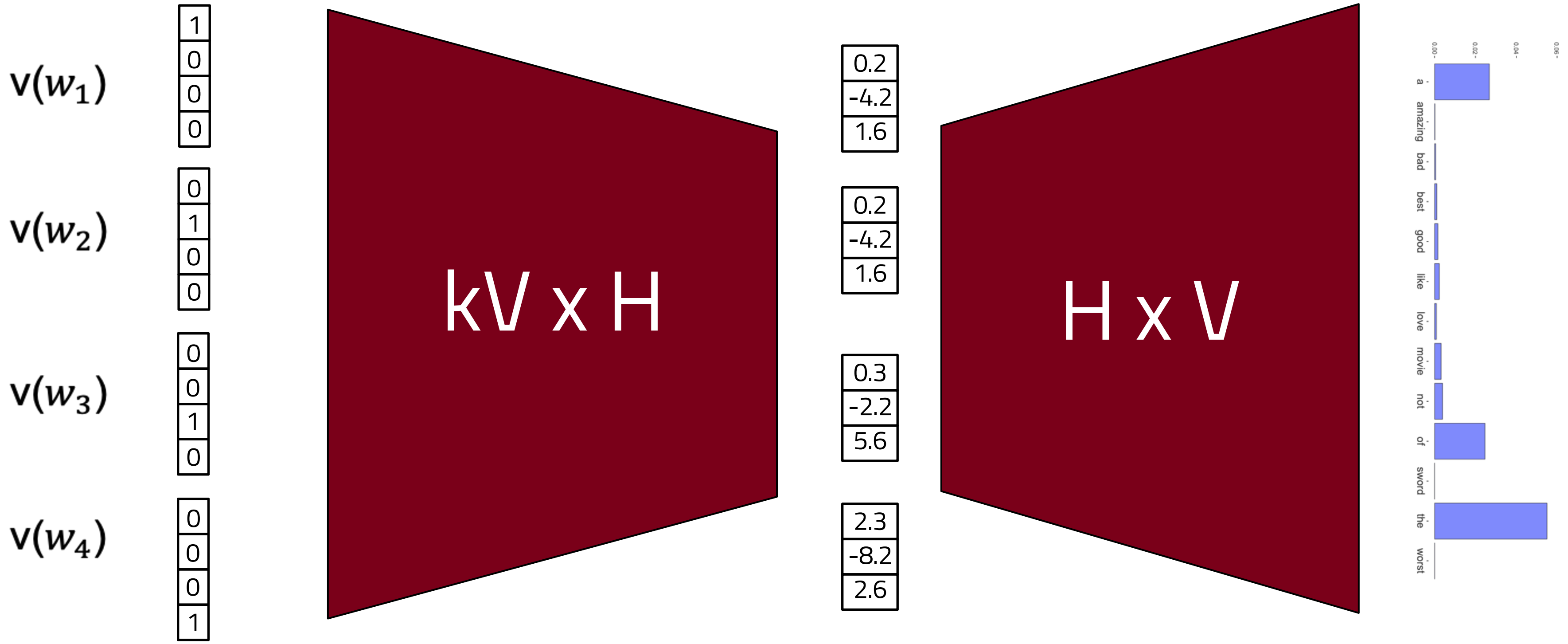
w_1 = tried

w_2 = to

w_3 = prepare

w_4 = midterms

Simple feed-forward multilayer perceptron
(e.g., one hidden layer)



One-hot encoding

Distributed representation

Multi-class (Vocab) classification

Bengio et al. 2003, A Neural Probabilistic Language Model

Neural LM

$$P(w) = P(w_i | w_{i-k} \ldots w_{i-1}) = softmax\,(W \cdot \boldsymbol{h})$$

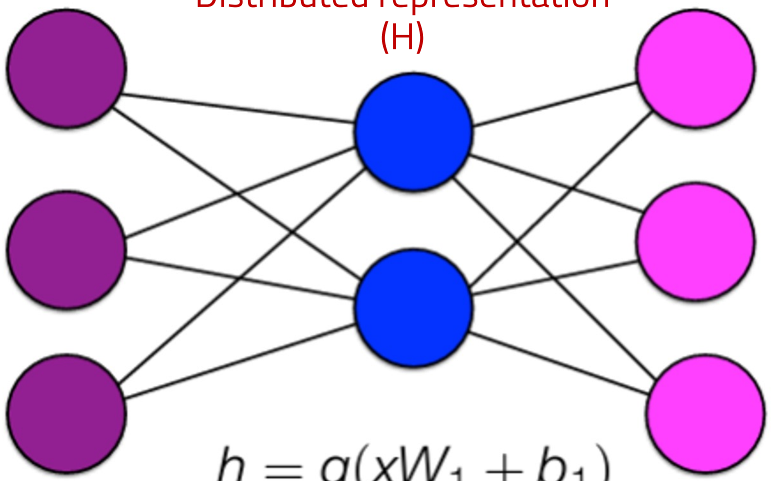
$W_1 \in \mathbb{R}^{kV \times H}$ $\quad$ $W_2 \in \mathbb{R}^{H \times V}$

$b_1 \in \mathbb{R}^{H}$ $\quad$ $b_2 \in \mathbb{R}^{V}$

One-hot encoding
(|x| = V)

Output space: |y| = V

Distributed representation
(H)



$$h = g(xW_1 + b_1)$$

$$x = [v(w_1); \ldots; v(w_k)]$$

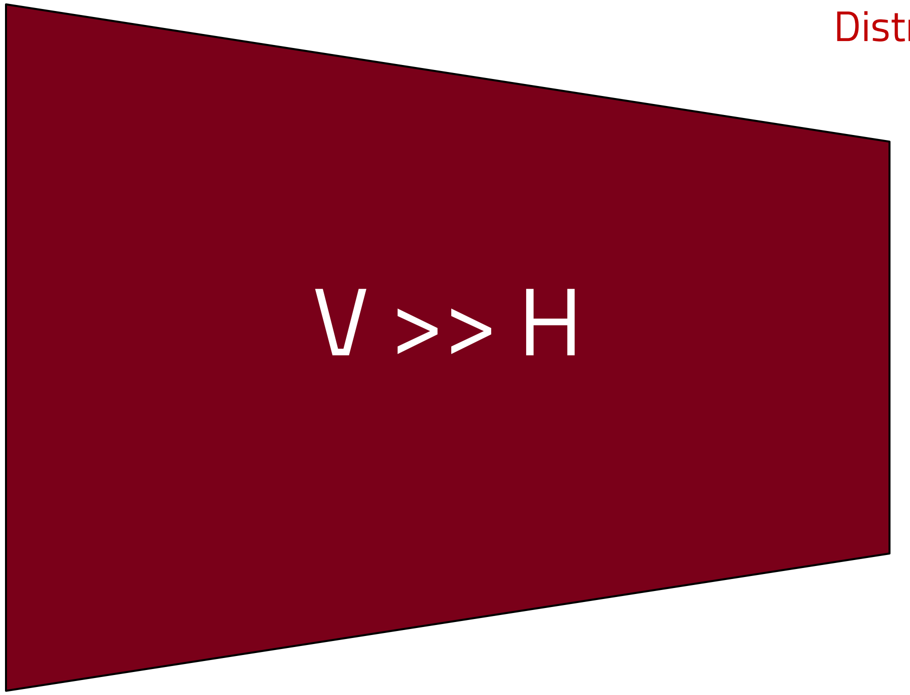
$$\hat{y} = \text{softmax}(hW_2 + b_2)$$

Bengio et al. 2003, A Neural Probabilistic Language Model

Neural LM

Represent high-dimensional words (and contexts) as low-dimensional vectors

One-hot encoding
(|x| = V)



Distributed representation
(|y| = H)

Bengio et al. 2003, A Neural Probabilistic Language Model

Conditioning context (X [k x V])

tried to prepare midterm but I was too tired of...

Next word to predict (Y)

Context window size: k=4

Conditioning context (X [k x V])

tried to prepare midterm but I was too tired of...

Next word to predict (Y)

Context window size: k=4

Conditioning context (X [k x V])

tried to prepare midterm but I was too tired of…

Next word to predict (Y)

Context window size: k=4

Neural LM against Ngram LM

Pros

- No sparsity problem
- Don't need to store all observed n-gram counts

Cons

- Fixed context window is too small (larger window, larger W)
  - Windows can never be large enough
- Different words are multiplied by completely different weights (W); no symmetry in how the inputs are processed.