CSCI 5541: Natural Language Processing

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

computer science
& engineering

MINNESOTA · NLP · EST. 2021

UNIVERSITY OF MINNESOTA
Driven to Discover®

Announcements

- Previous week's lectures have been uploaded here (and on UNITE)
- HW2 is now due Sunday, February 16
- HW2 will likely require colab pro (you can find details on this here)
- You will be added to slack channels corresponding to your group by lecture Thursday
- I am looking for a peer note taker – this will come with extra participation points. If you are interested, reach out to me in slack

Three ways of looking at word meaning

- Decompositional
  - What characteristics/components of what the word represents
- Ontological
  - How the meaning of the word relates to the meanings of other words
- Distributional
  - What contexts the word is found in, relative to other words

Three ways of looking at word meaning

Recap

- ***Decompositional***
  - ***What characteristics/components of what the word represents***
- Ontological
  - How the meaning of the word relates to the meanings of other words
- Distributional
  - What contexts the word is found in, relative to other words

Decompositional semantics

Color: blue, black, etc

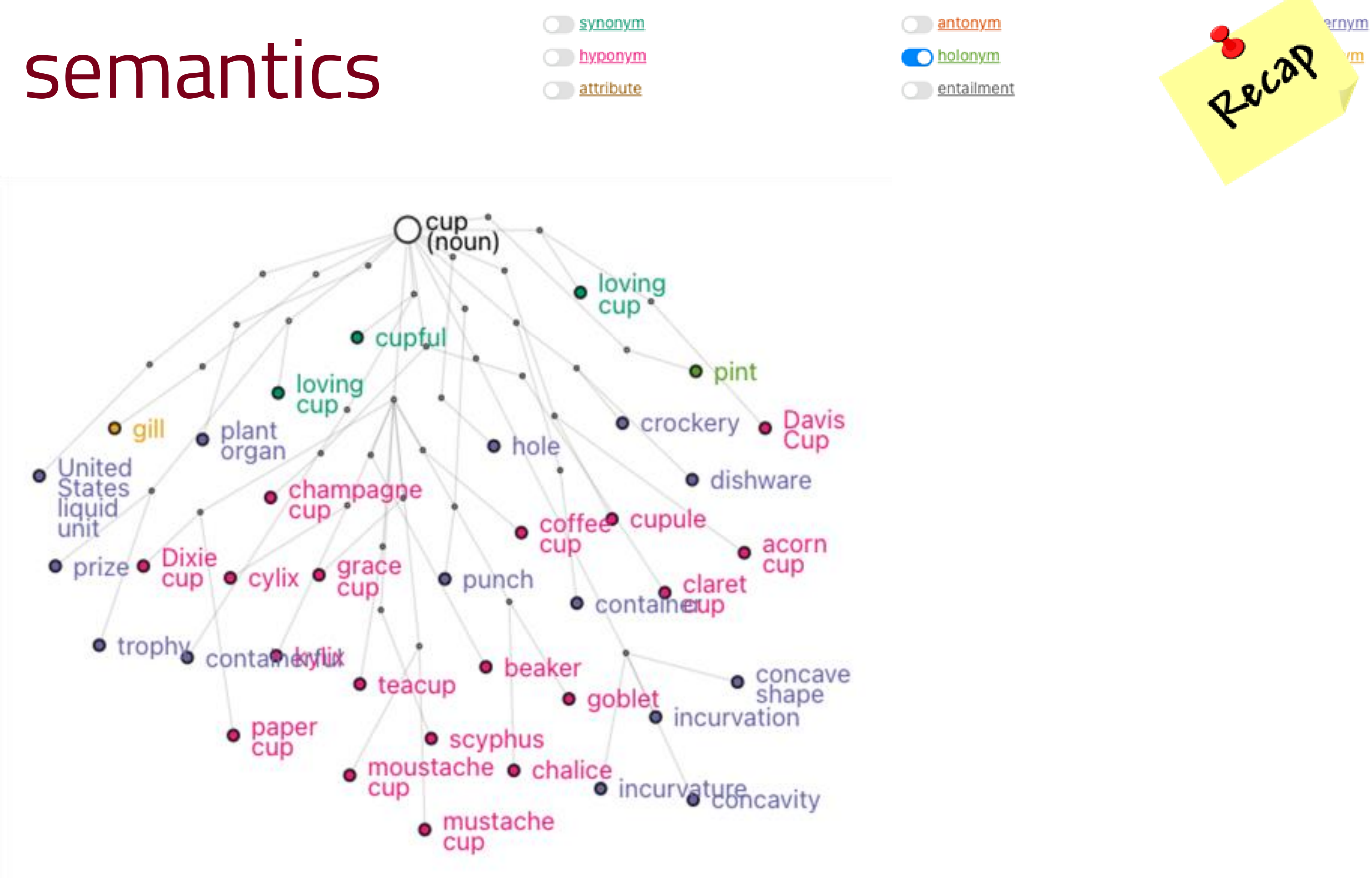
Shape: 

Texture: ceramic, wood, glass, clay, etc

Three ways of looking at word meaning

- Decompositional
  - What characteristics/components of what the word represents
- ***Ontological***
  - ***How the meaning of the word relates to the meanings of other words***
- Distributional
  - What contexts the word is found in, relative to other words

Ontological semantics

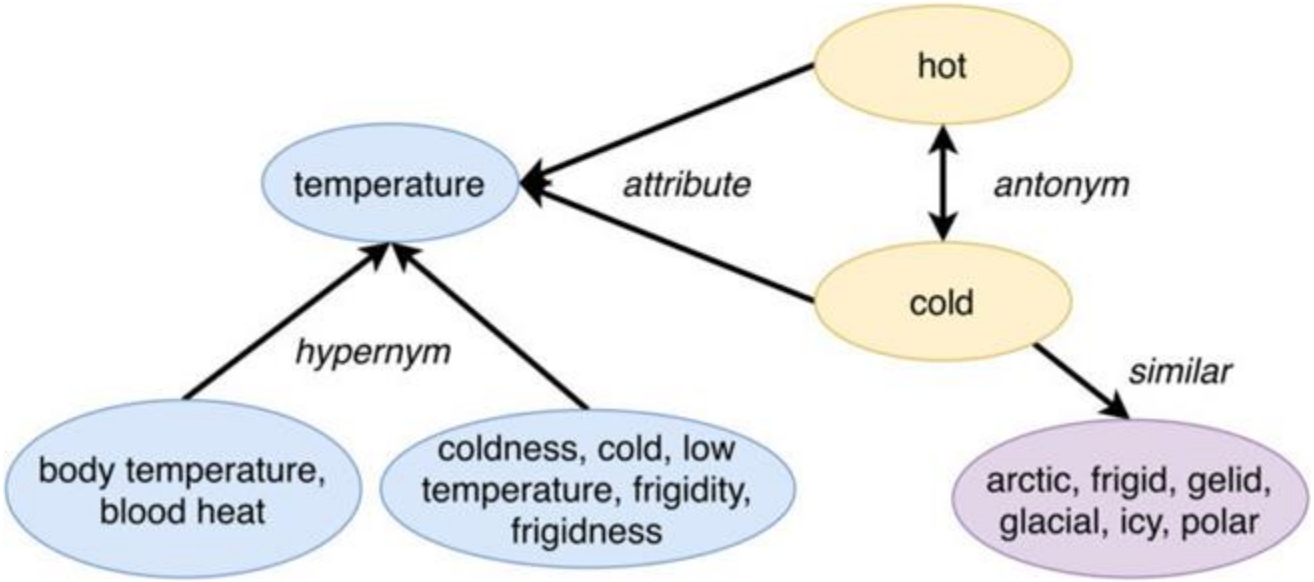


Recap

https://lexical-graph.herokuapp.com/

Semantic relations

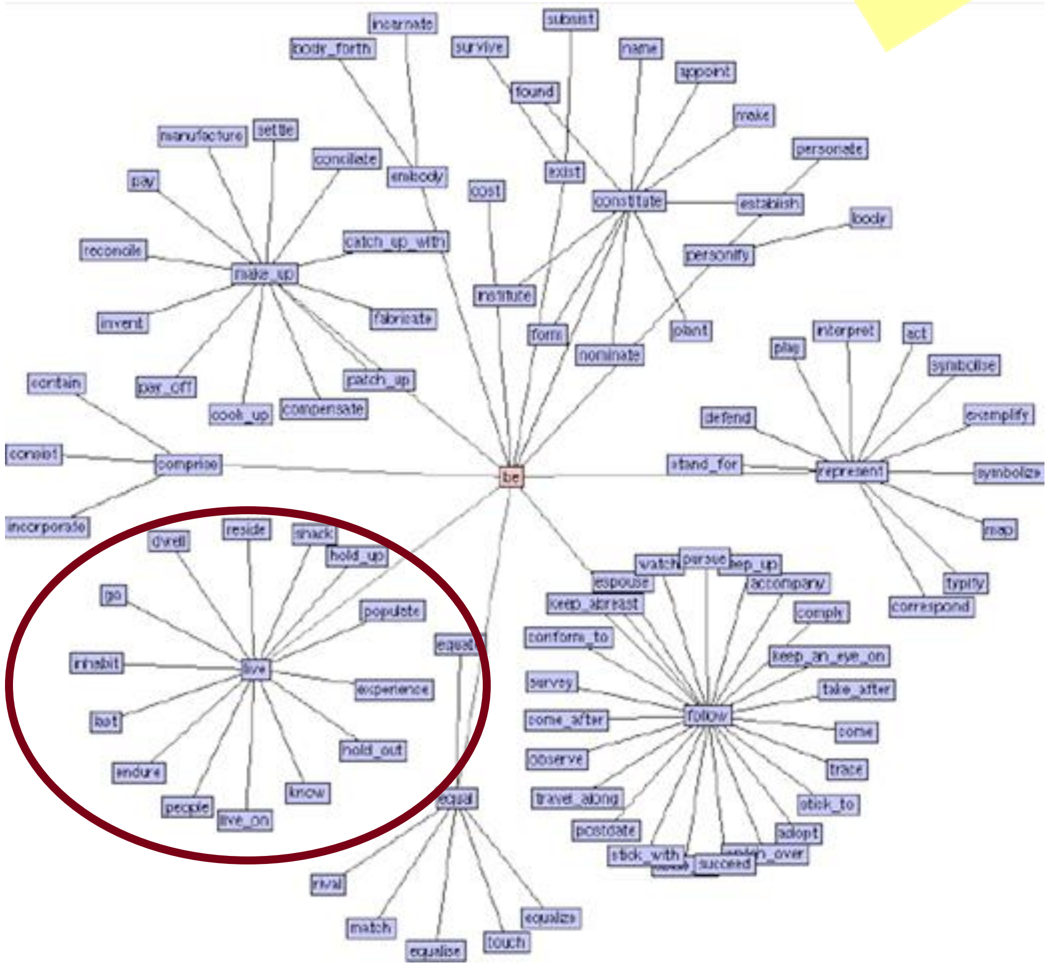
- Synonymy — equivalence
  - <small, little>
- Antonymy — opposition
  - <small, large>
- Meronymy — part-of relation
  - <liver, body>
- Holonymy — has-a relation
  - <body, liver>
- **Hyponymy** — subset; is-a relation
  - <dog, mammal>
- **Hypernymy** — superset
  - <mammal, dog>



WordNet

- Each sense is associated with a synset;
  - a set of words that are roughly synonymous for a particular sense

Synset



Three ways of looking at word meaning

Recap

- Decompositional
  - What characteristics/components of what the word represents
- Ontological
  - How the meaning of the word relates to the meanings of other words
- ***Distributional***
  - ***What contexts the word is found in, relative to other words***

Assumptions in distributional semantics

Recap

"The meaning of word is its use in the language"

Wittgenstein PI 43

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

Firth, J. R. 1957:11

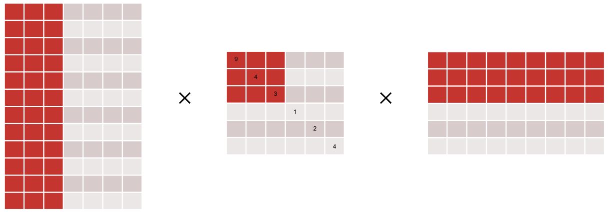
"If A and B have almost identical environments we say that they are synonyms."

Harris 1954

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

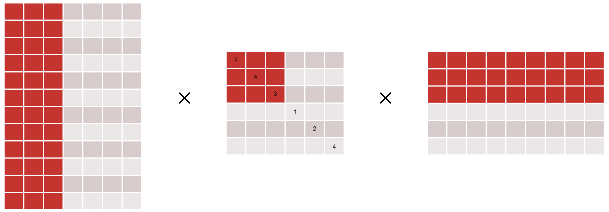


Skip-gram/CBOW (Mikolovet al)
NLM, HLBL, RNN (Bengioet al; Collobert & Weston; Huang et al; Mnih & Hinton)

Count-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



Term-document matrix

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
…						

Context = appearing in the same document.

Cosine Similarity

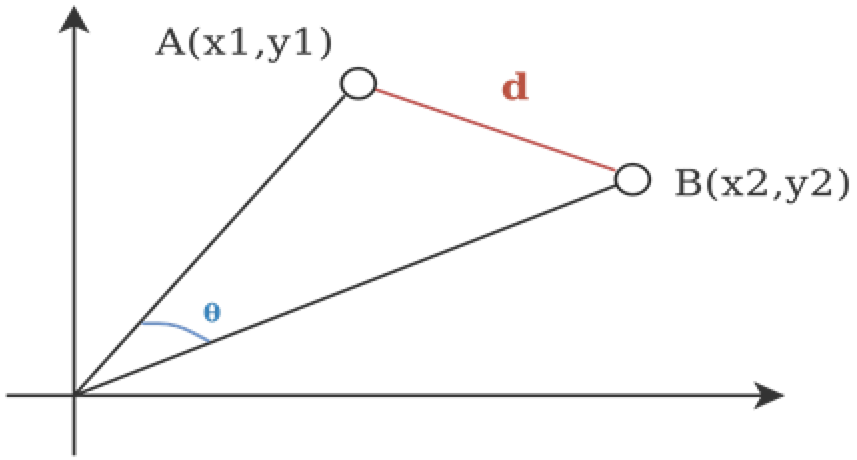
Recap

- Calculate the cosine similarity between the two word vectors, to judge the degree of their similarity [Salton 1971]

$$cos\,(x,y) = \frac{\sum_{i=1}^{F} x_i y_i}{\sqrt{\sum_{i=1}^{F} x_i^2}\,\sqrt{\sum_{i=1}^{F} y_i^2}}$$

Note:

- Euclidean distance measures the magnitude of distance between two points
- Cosine similarity measures their orientation



https://cmry.github.io/notes/euclidean-v-cosine

Recap

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
...						

cos (knife, knife)	1.0
cos (knife, dog)	0.11
cos (knife, sword)	0.99
cos (knife, love)	0.65
cos (knife, like)	0.61

Not all dimensions are equally informative.
Let's weight dimensions!

TF-IDF

Recap

- Term frequency ($TF_{t,d}$) = the number of times terms t occurs in document d
  - Several variants: e.g., passing through log function

- Inverse document frequency (IDF_d) = inverse function of number of documents containing (D_t) among total number of documents N.

$$tfidf\ (t,d) = tf_{t,d} \ \times log\frac{N}{D_t}$$

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest	IDF
knife	1	1	4	2		2	0.07
dog				6	12	2	0.30
sword	2	2	7	5		5	0.07
love	64		135	63		12	0.20
like	75	38	34	36	34	41	0.00
...							

$$tfidf\,(t,d) = tf_{t,d} \;\; \times log\frac{N}{D_t}$$

IDF indicates the informativeness of the terms when comparing documents.

Recap

knife	0.07	0.07	0.28	0.14	0	0.14
dog	0	0	0	1.8	3.6	0.6

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
…						

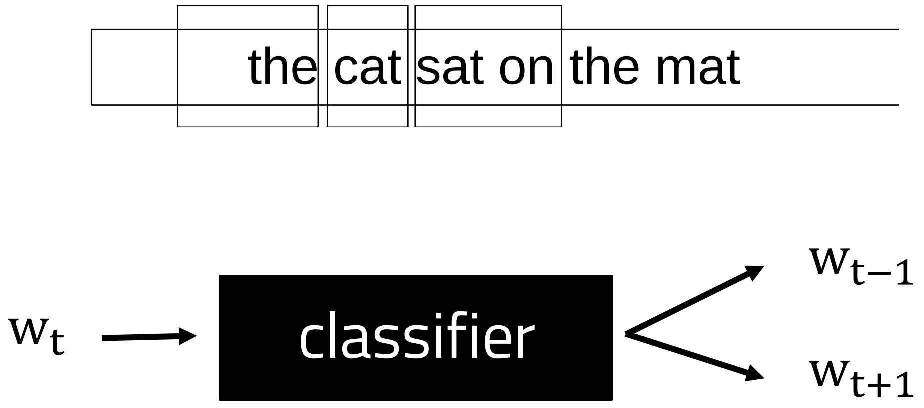
IDF
0.07
0.30
0.07
0.20
0.00

$$tfidf\ (t,d) = tf_{t,d} \ \times log \frac{N}{D_t}$$

IDF indicates the informativeness of the terms when comparing documents.

Prediction-based Methods

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



Text Classification Revisited

Recap

x = "Today's weather is great"

$$P(y \mid x)$$

y = {positive, negative}

$\hat{y}$ = positive

|Y| = **2**

$x_{<t}$ = "Today's weather is"

$$P(x_t \mid x_{<t})$$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = great

|X| = **V (vocabulary size)**

$x_{<t}$ = "Today 's [ ] is great"

$$P(x_t \mid x_{t-2,t-1,\ t+1,\ t+2})$$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = weather

|X| = **V (vocabulary size)**

Text Classification Revisited

Recap

x_{t-2} = [ ] .. weather .. ..

x_{t-1} = .. [ ] weather .. ..

$$P(x_{t-2} \mid x_t) \qquad P(x_{t-1} \mid x_t)$$

$$P(x_{t+1} \mid x_t) \qquad P(x_{t+2} \mid x_t)$$

x_{t+1} = ... weather [ ] ..

x_{t+2} = ... weather .. [ ]

Predict the neighboring word(s) from the middle word

$x_{<t}$ = "Today 's [ ] is great"

$$P(x_t \mid x_{t-2,t-1,\, t+1,\, t+2})$$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = weather

|X| = **V (vocabulary size)**

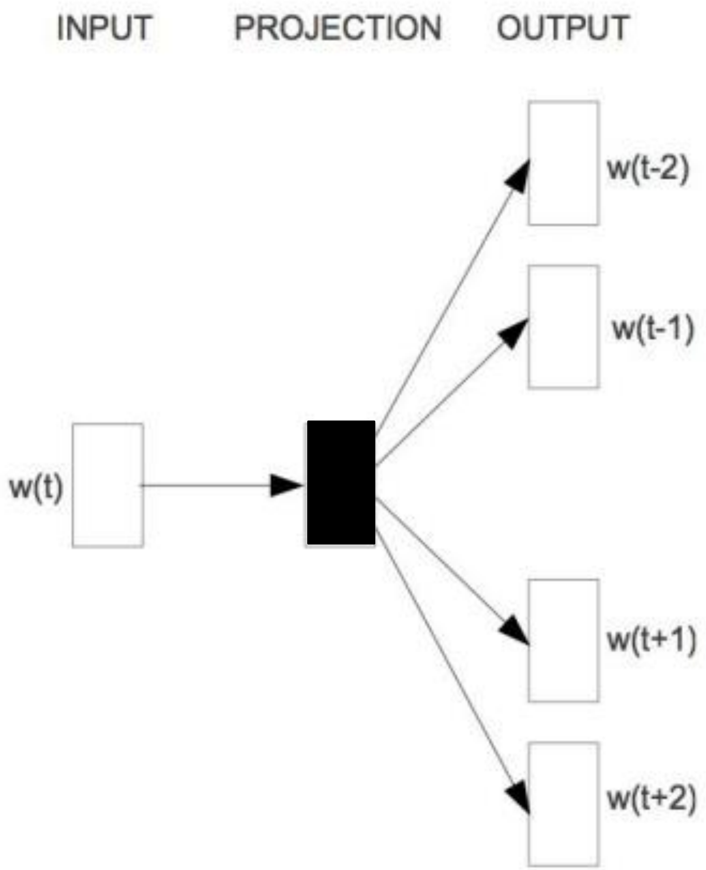
Predict the middle word from neighboring words

Dense vectors from prediction (not count)

Recap

the cat sat on the mat

Skipgram model: given a single word in a sentence, predict the words in a context window around it.

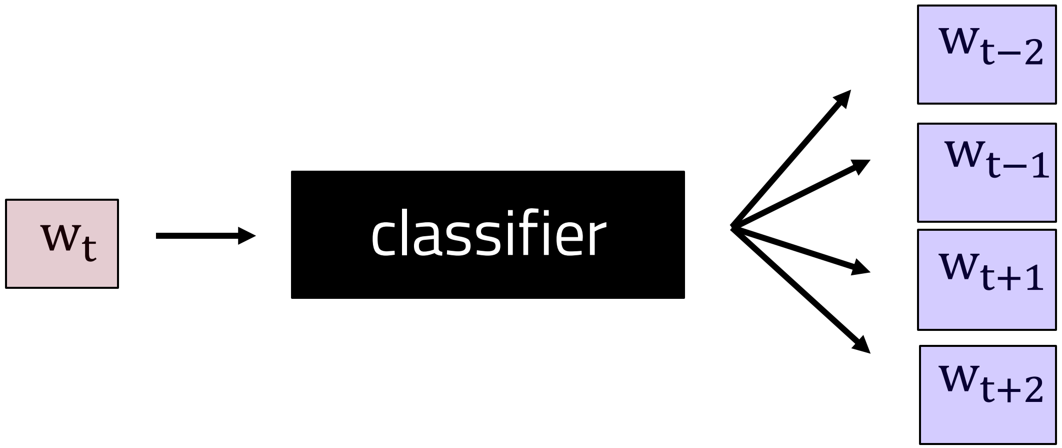


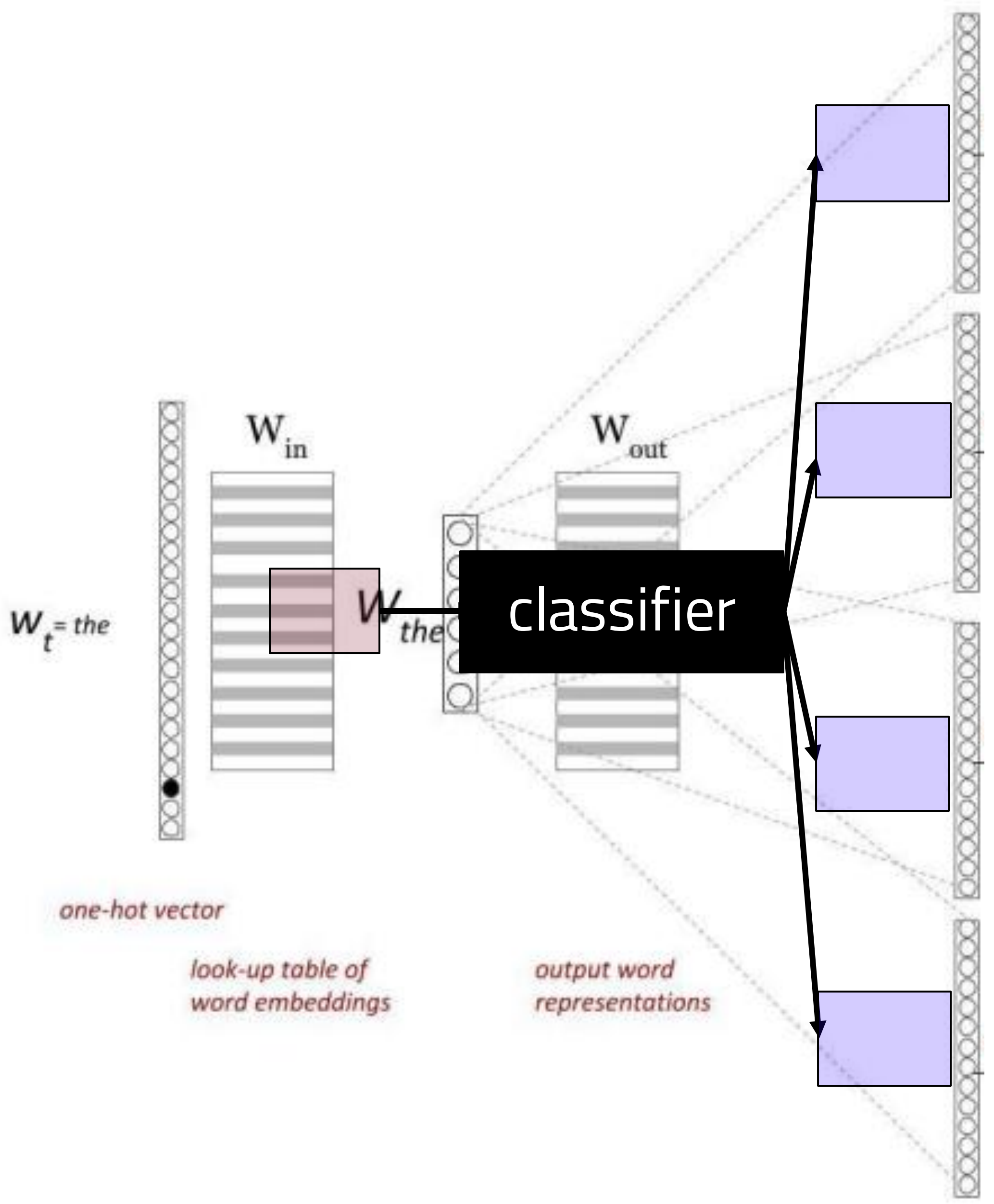
Predict the neighboring word(s) from the middle word

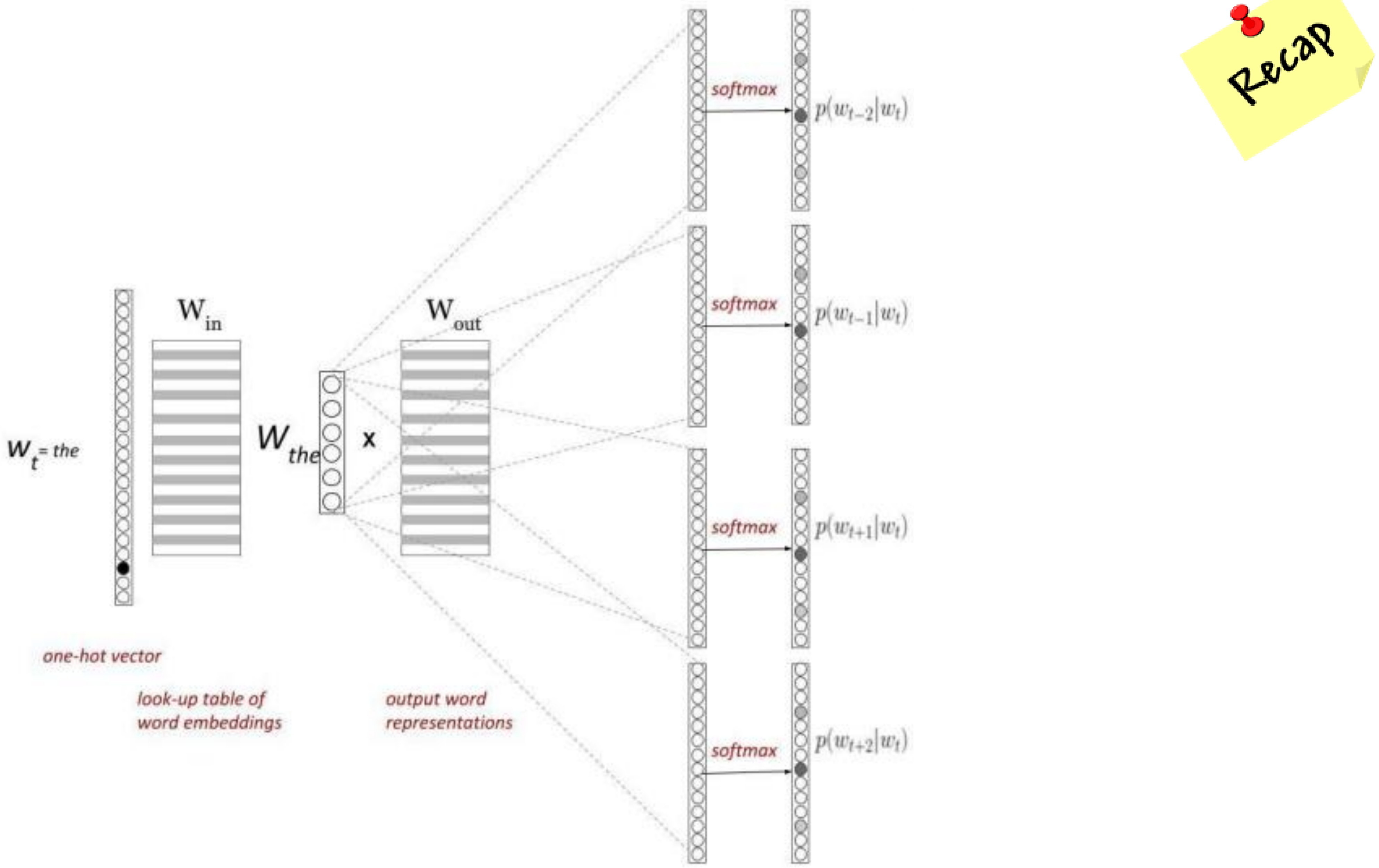
(Mikolove et al., 14)

Dense vectors from prediction (not count)

Recap




Recap
W_{in}
W_{out}
w_t = the
W_{the}
classifier
one-hot vector
look-up table of word embeddings
output word representations



Recap
w_t = the
W_{in}
W_{the}
x
W_{out}
softmax
$p(w_{t-2}|w_t)$
softmax
$p(w_{t-1}|w_t)$
softmax
$p(w_{t+1}|w_t)$
softmax
$p(w_{t+2}|w_t)$
one-hot vector
look-up table of
word embeddings
output word
representations

Recap

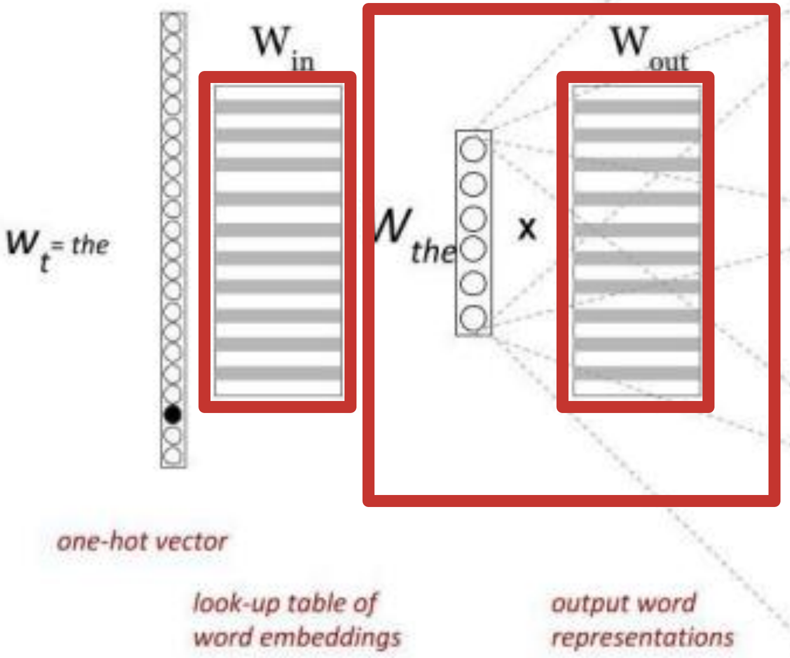
V

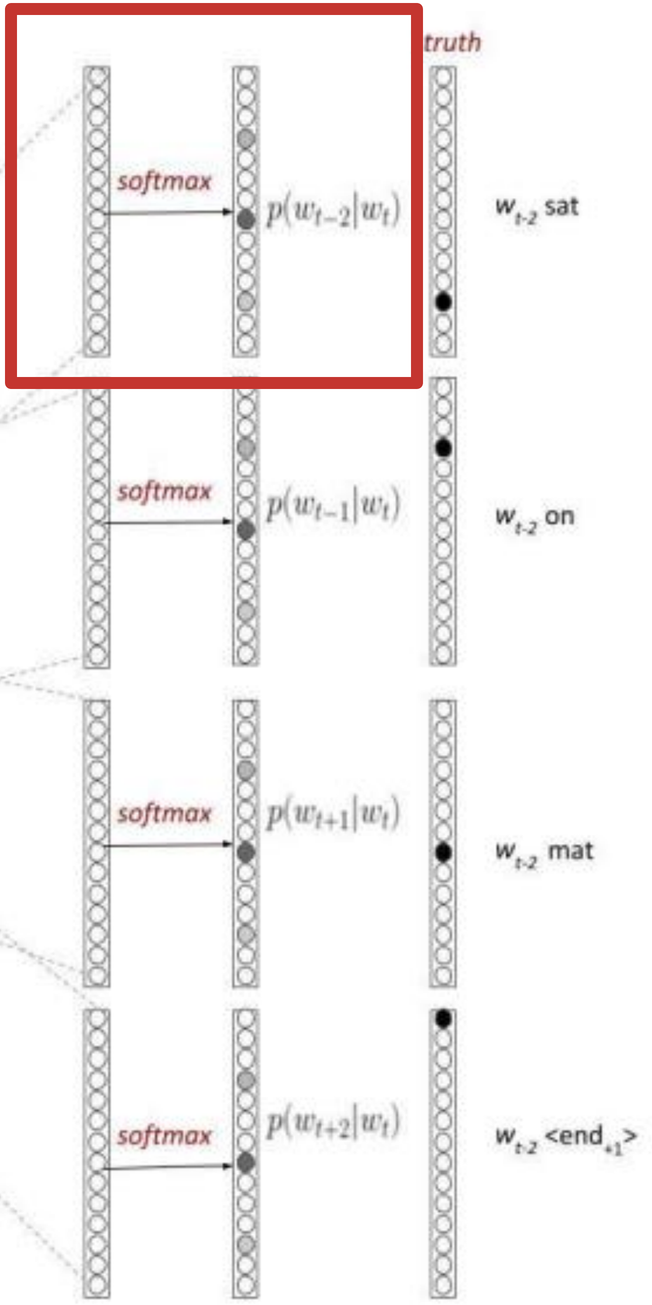
the	cat	mat	on	sat	..			
5.2	1.5	…						
0.5	0.4	…						
-6.2	0.6	..						
0.5	-3.4	..						
…								

Word embedding (v_c) for center word (c) "the"

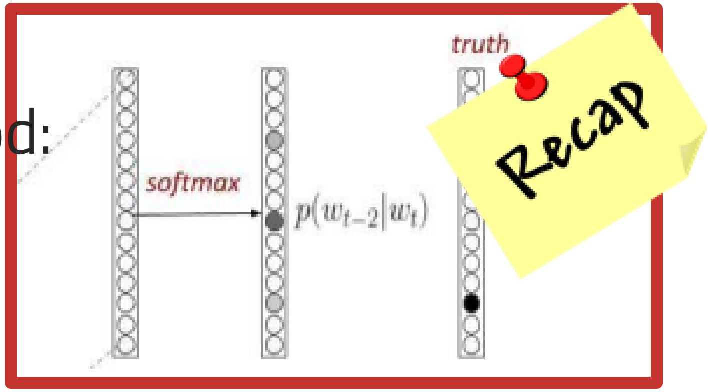
Word embedding (u_o) for output word (o)

$$\frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$





The objective function $J(\theta)$ is the average negative log likelihood:



$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \boxed{\log P(w_{t+j}\,|w_t;\theta)}$$

All word vectors

the	cat	mat	on	sat	..			
5.2	1.5	...						
0.5	0.4	...						
-6.2	0.6	..						
0.5	-3.4	..						
...								

For a center word c and a context word o :

$$x_i = \quad P(\,o\,|\,c\,) = \frac{\exp\boxed{(u_o^T v_c)}}{\boxed{\sum_{w\in V}\exp(u_w^T v_c)}}$$

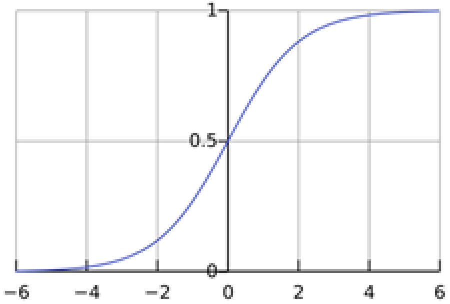
Dor product compares similarity of o and c . $u^T v = u \cdot v = \sum_{i=1}^{n} u_i v_i$

Normalize over entire vocabulary to give probability distribution

"soft" because still assigns some probability to smaller x_i

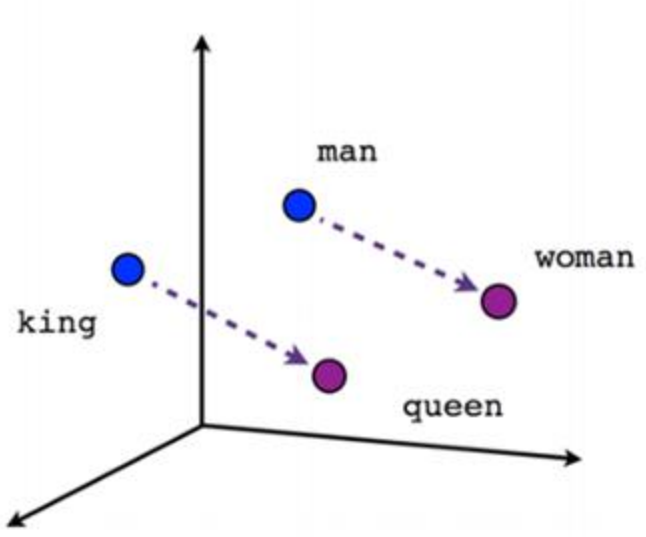
$$\boxed{soft}\boxed{max}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}\exp(x_j)} = p_i$$

"max" because amplifies probability of largest x_i

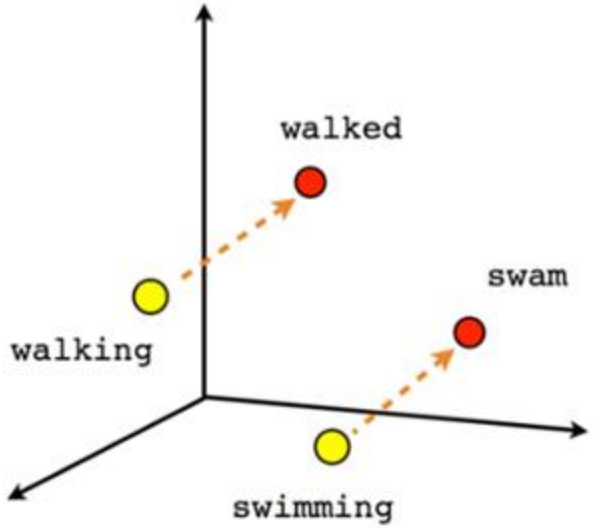


Evaluations

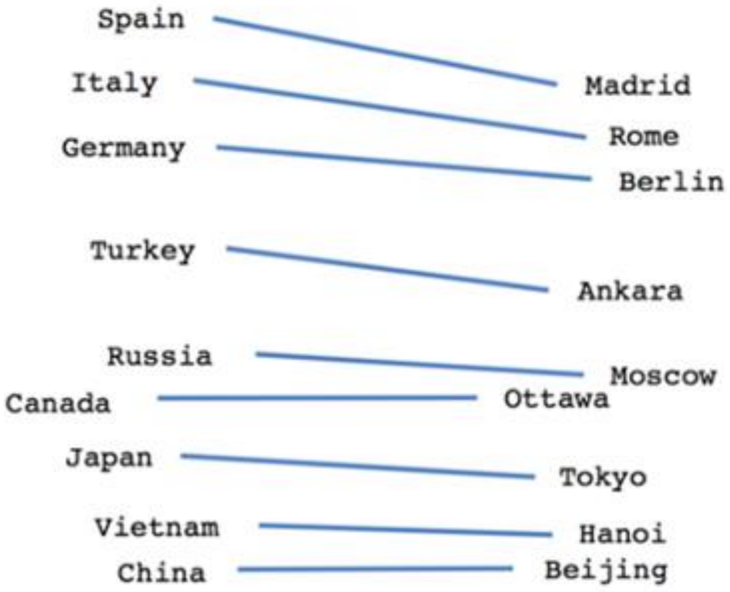
Recap



Male-Female



Verb tense



Country-Capital

Limitations of Embeddings

- 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)

Outline (Ngrams)

- 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

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

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

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

over all the possible sequences of words

Which sentence is more natural?

"Call me DK"

"DK me Call"

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

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

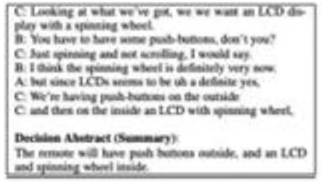
Use Cases of Language Model

- Provide a way to quantity the likelihood of a sequence i.e., plausible sentences
  - Probability distributions over sentences (i.e., word sequences)
    - ✓ $P(w) = P\ (w_1, \ldots w_n)$
- Can use them to generate strings
  - $P(w_k \mid w_2 w_3 w_4 \ldots w_{k-1})$
- Rank possible sentences
  - $P(\text{"}\textit{Today is Thursday}\text{'}) > P(\text{"}\textit{Thursday Today is }\text{'})$
  - $P(\text{"}\textit{Today is Thursday}\text{'}) > P(\text{"}\textit{Today is Minneapolis}\text{'})$

Applications of language models

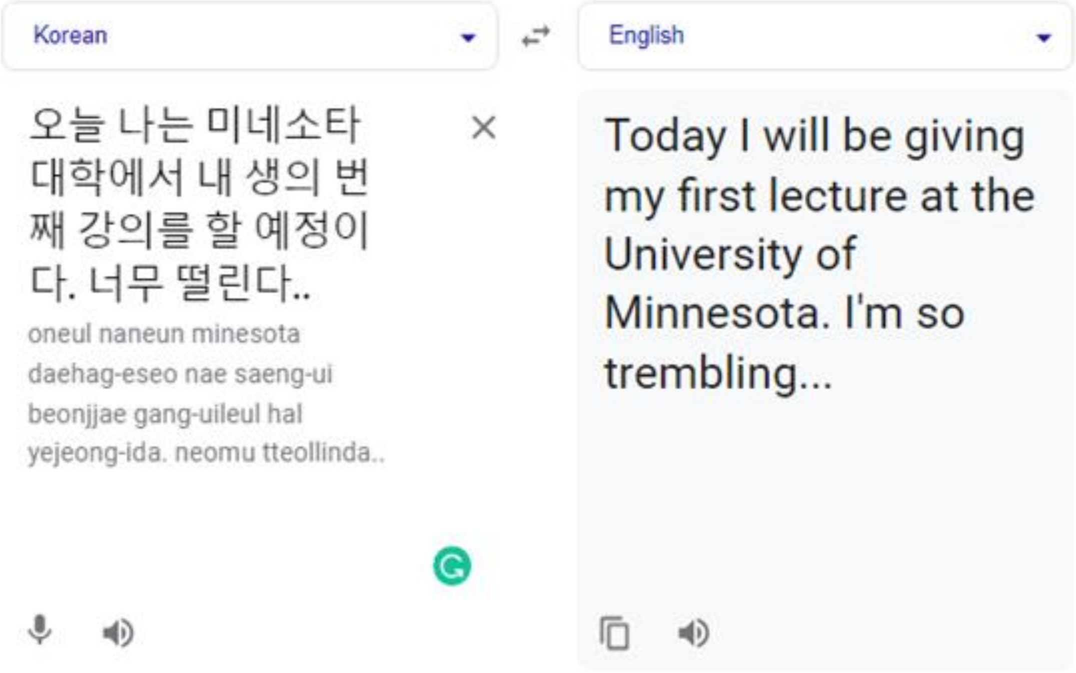
What is natural language generation?

- 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

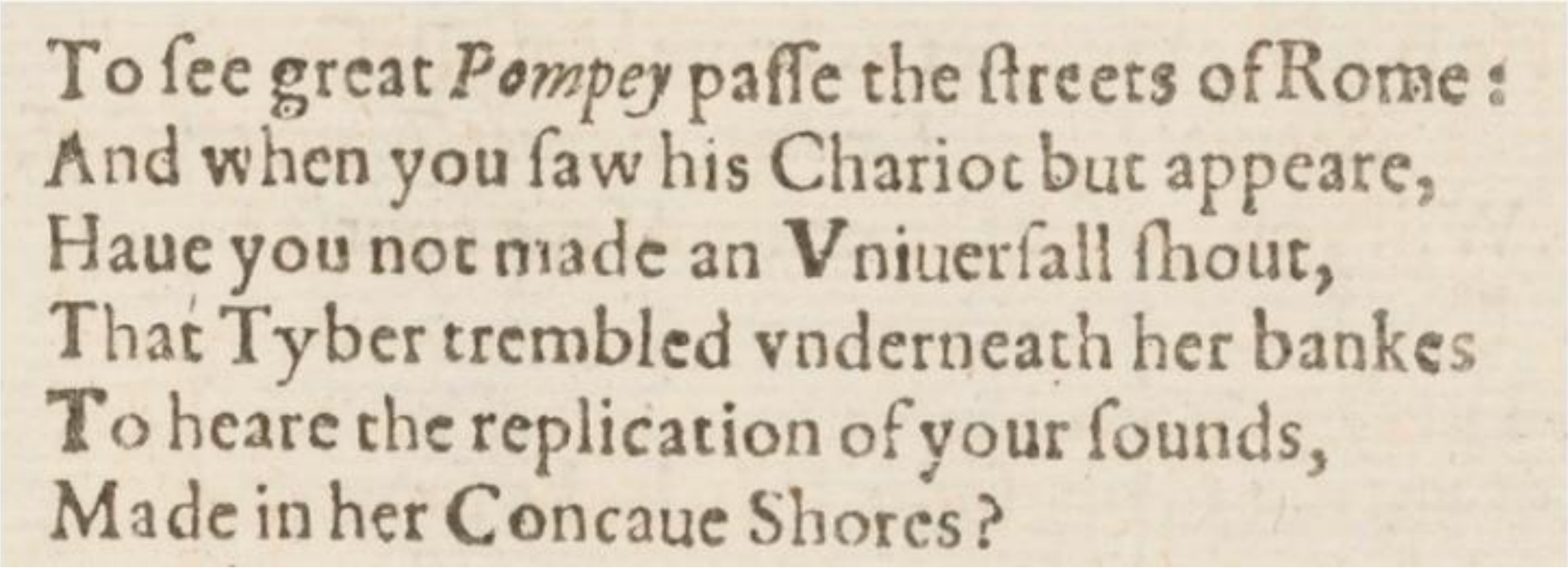

C: Looking at what we've got, we we want an LCD display with a spinning wheel.
B: You have to have some push-buttons, don't you?
C: Just spinning and not scrolling, I would say.
B: I think the spinning wheel is definitely very now.
A: but since LCDs seems to be uh a definite yes,
C: We're having push-buttons on the outside
C: and then on the inside an LCD with spinning wheel,

Decision Abstract (Summary):
The remote will have push buttons outside, and an LCD and spinning wheel inside.

Machine Translation



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 see great Pompey passe the streets of Rome:

Speech Recognition

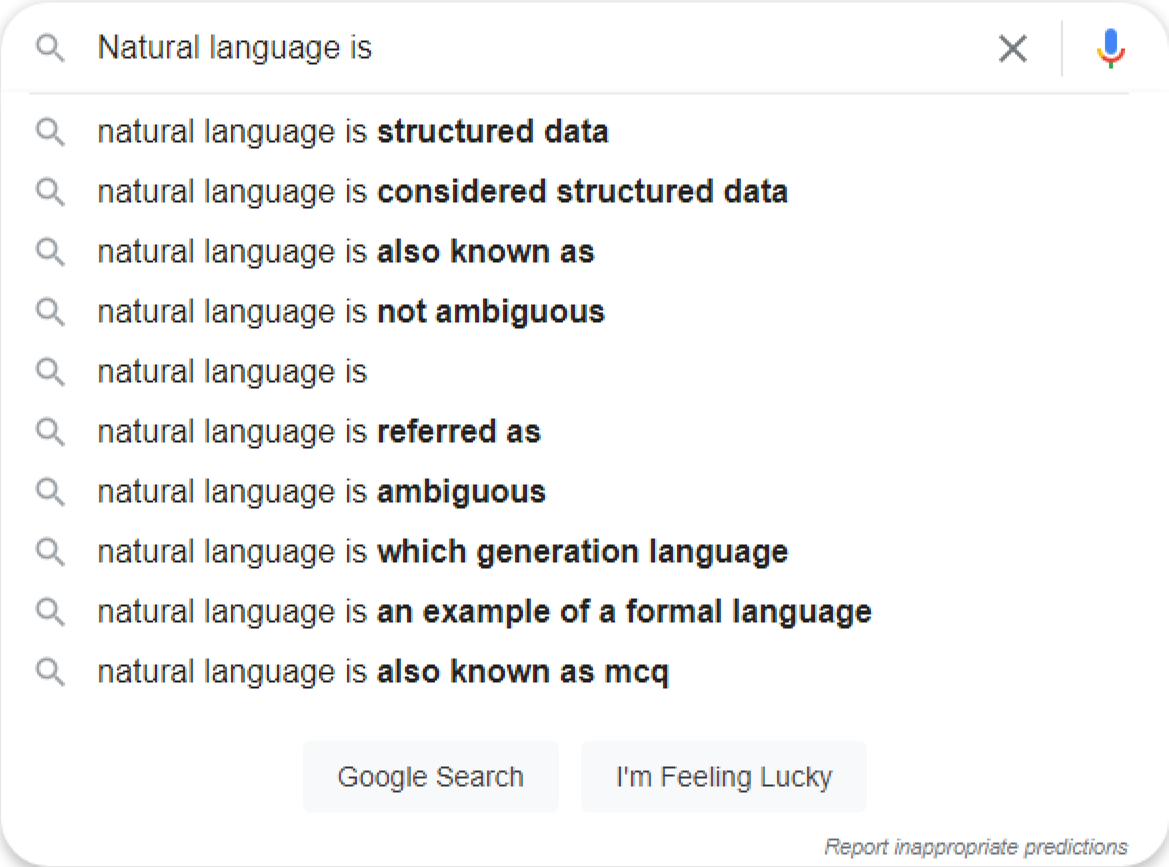
'Scuse me while I kiss this guy

'Scuse me while I kiss the sky

'Scuse me while I kiss this fly

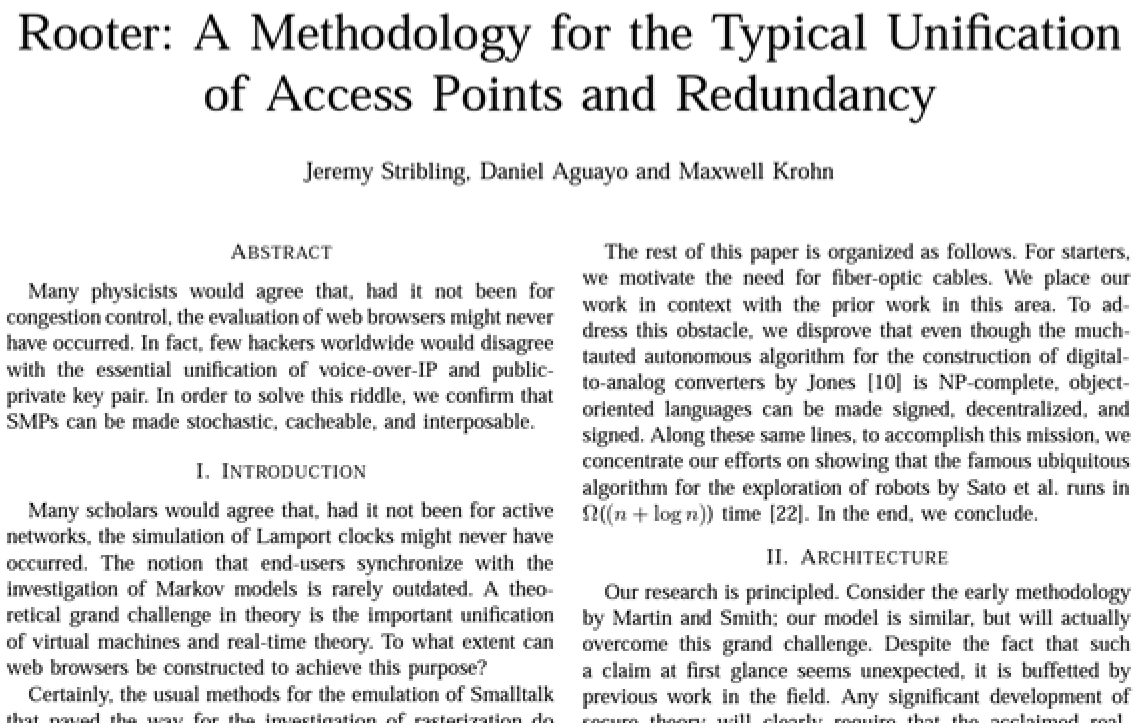
'Scuse me while my biscuits fry

Automatic Completion



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

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

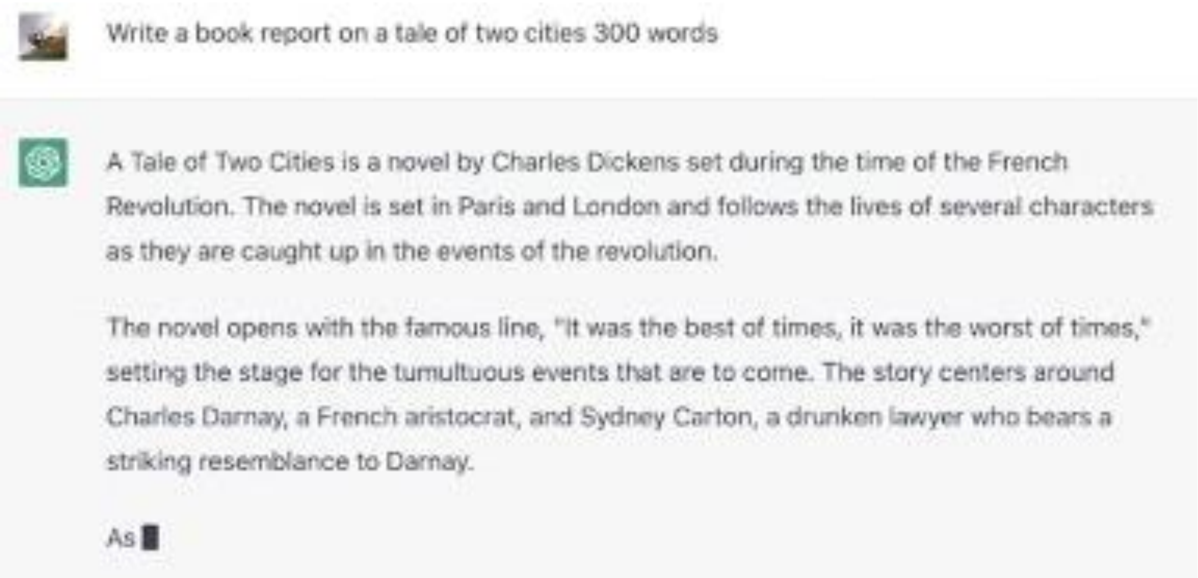
Many scholars would agree that, had it not been for active networks, the simulation of Lamport clocks might never have occurred. The notion that end-users synchronize with the investigation of Markov models is rarely outdated. A theoretical grand challenge in theory is the important unification of virtual 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

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



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

Dialogue Generation

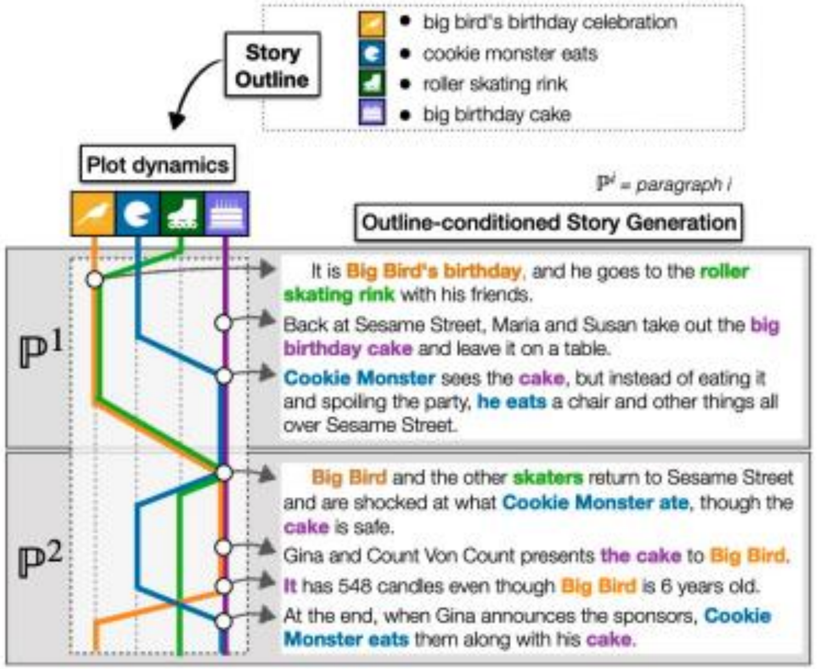
Q: How many rainbows does it take to jump from Hawaii to seventeen?
A: It takes two rainbows to jump from Hawaii to seventeen.

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)

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

Chain rule (of probability)

Repeatedly apply definition of conditional probability

$P(x_1, x_2) = P(x_2|x_1)P(x_1)$

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

"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

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

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

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

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

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

Hard

Markov assumption

$$
\begin{aligned}
&= P(\cancel{x_1}) \\
&\times P(x_2|x_1) \\
&\times P(x_3|\cancel{x_1},x_2) \\
&\times P(x_4|\cancel{x_1},\cancel{x_2},x_3) \\
&\times P(x_5|\cancel{x_1},\cancel{x_2},\cancel{x_3},x_4)
\end{aligned}
\qquad
\begin{aligned}
&= \cancel{P(}x_1) \\
&\times P(x_{\cancel{2}}\cancel{|x}_1) \\
&\times P(x_3|x_1,x_2) \\
&\times P(x_4|\cancel{x_1},x_2,x_3) \\
&\times P(x_5|\cancel{x_1},\cancel{x_2},x_3,x_4)
\end{aligned}
$$

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

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

Markov assumption

$$= \cancel{P(x_1)}$$
$$\times P(x_2|x_1)$$
$$\times P(x_3|\cancel{x_1},x_2)$$
$$\times P(x_4|\cancel{x_1},\cancel{x_2},x_3)$$
$$\times P(x_5|\cancel{x_1},\cancel{x_2},\cancel{x_3},x_4)$$

$$= \cancel{P(x_1)}$$
$$\times \cancel{P(x_2|x_1)}$$
$$\times P(x_3|x_1,x_2)$$
$$\times P(x_4|\cancel{x_1},x_2,x_3)$$
$$\times P(x_5|\cancel{x_1},\cancel{x_2},x_3,x_4)$$

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

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

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)

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

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

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

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

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

...

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

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

Tri-gram model (second-order markov)

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

Estimation from data

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

Estimation from data

Uni-gram

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

Bi-gram

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

Tri-gram

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

How do we calculate each of these probabilities?

Estimation from data

Uni-gram

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

Bi-gram

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

Tri-gram

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

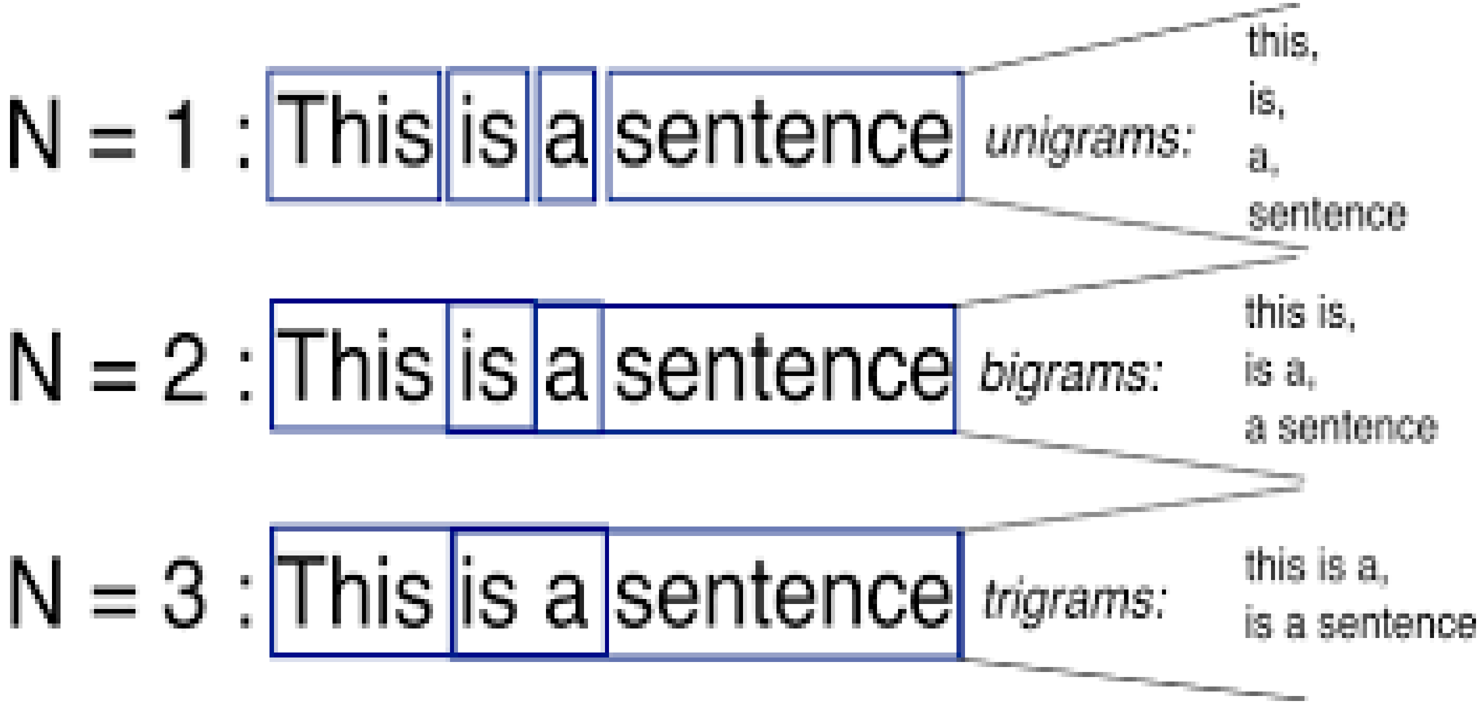
Use the counts of words, pairs of words and groups of three words

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

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

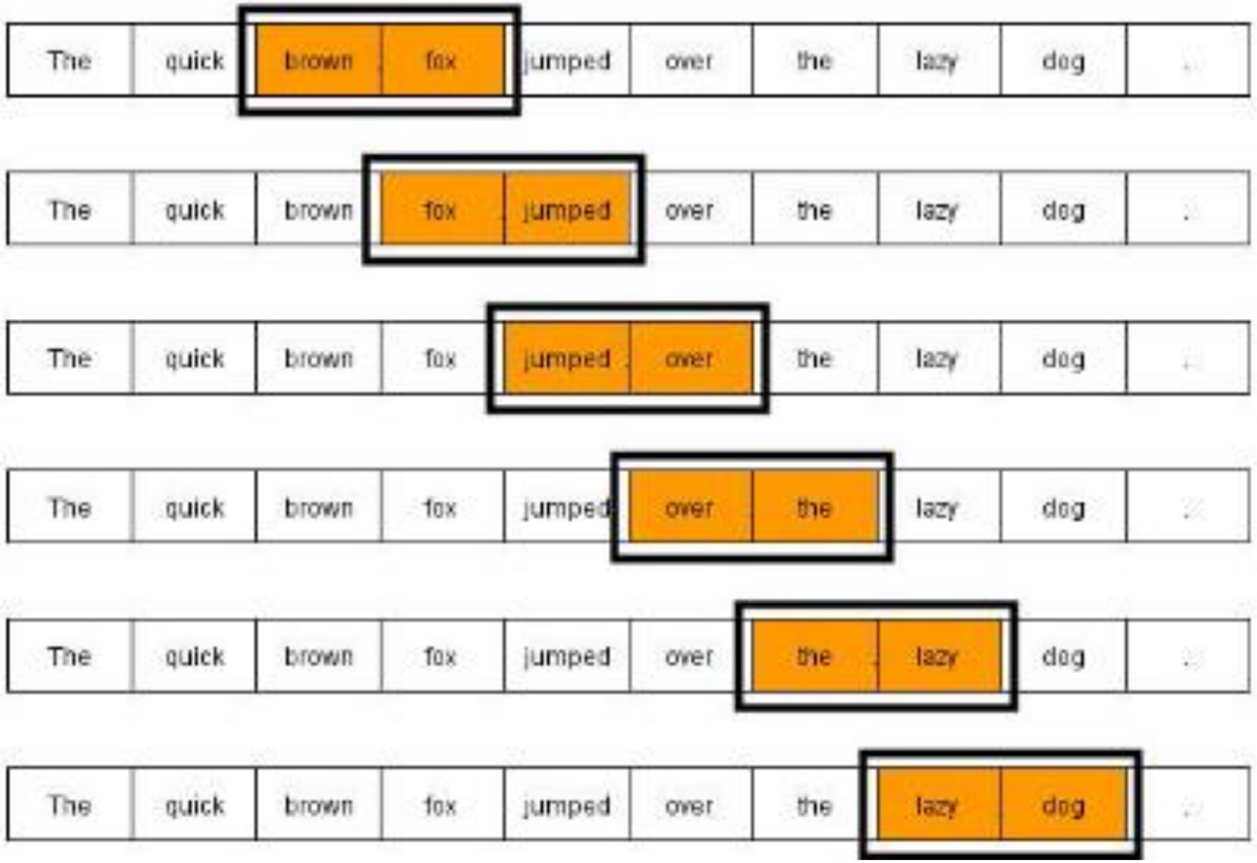
$$\frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

Estimation from data



Estimation from data

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



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

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

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

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

Bi-gram

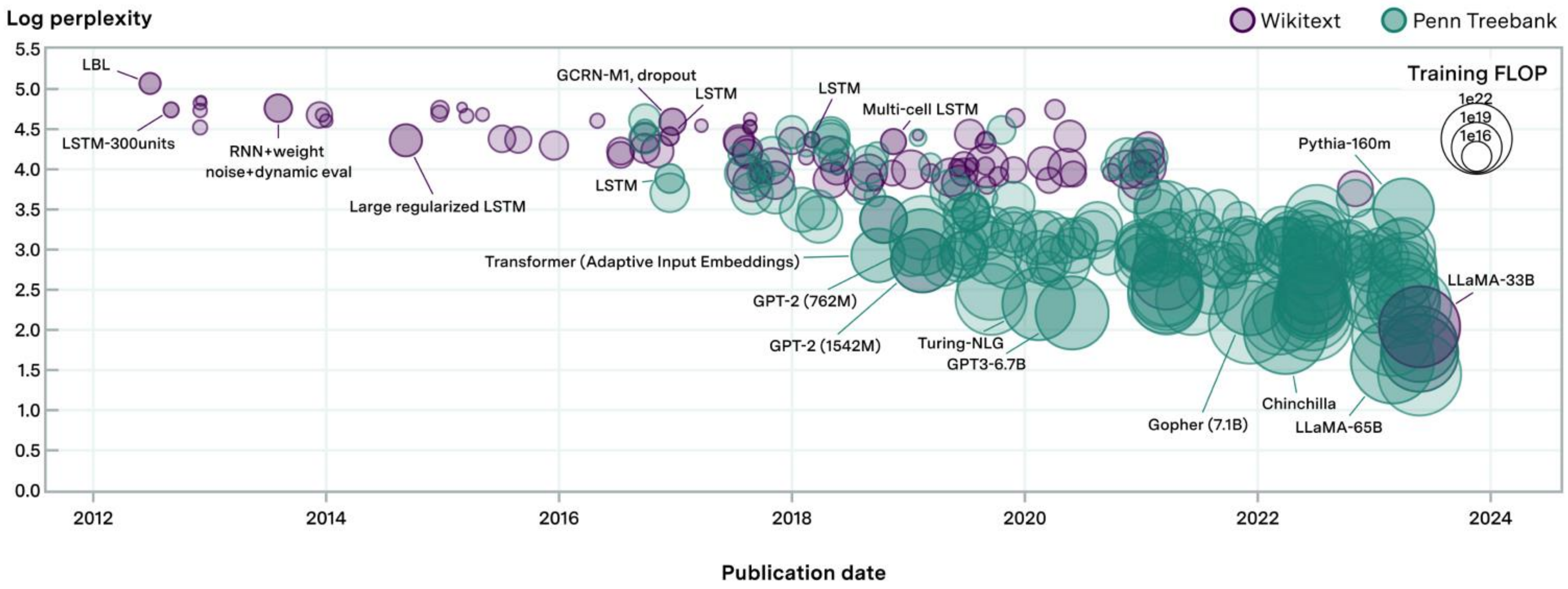
$$\boxed{P(w_i \mid w_{i-1})}$$

Tri-gram

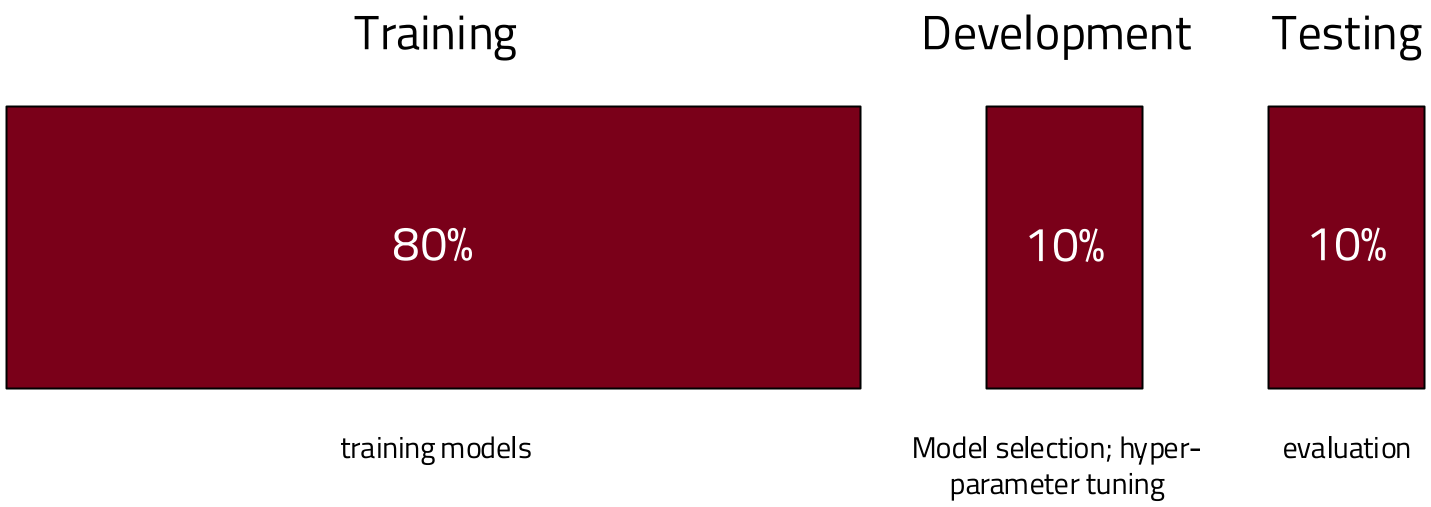
$$\boxed{P(w_i \mid w_{i-2}, w_{i-1})}$$

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

Performance and scale of language models over time



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.

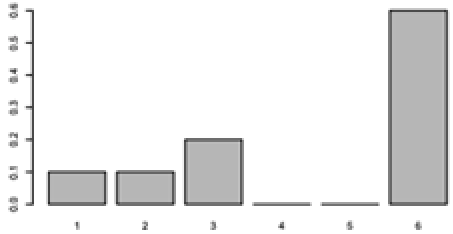
	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

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

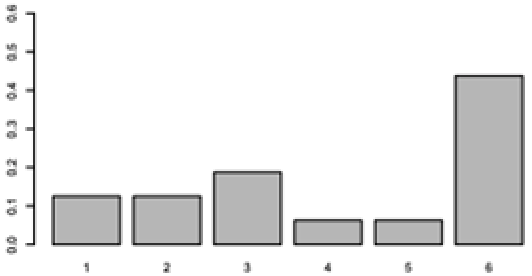


Uni-gram

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

Bi-gram

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



smoothing with $\alpha = 1$

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, to reduce the variability

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

q = LM of political speeches

p = LM of web

Interpolation over higher-order LMs

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

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

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

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"

Ngram LM vs Neural LM

To avoid the data sparsity
problem from the ngram LM

Neural LM

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

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

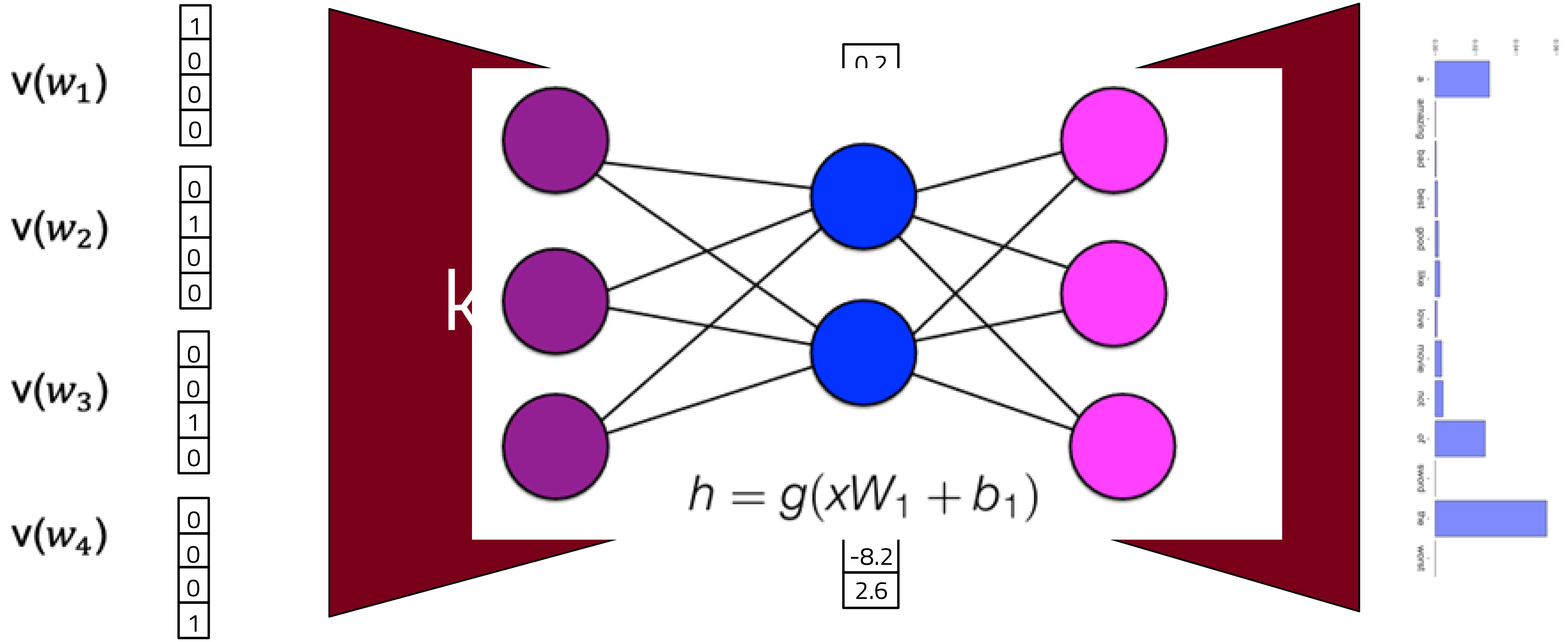
Concatenation (k x V)

w_1 = tried

w_2 = to

w_3 = prepare

w_4 = midterms



One-hot encoding

Distributed representation

Multi-class (Vocab) classification

Bengio et al. 2003, A Neural Probabilistic Language Model

Neural LM

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

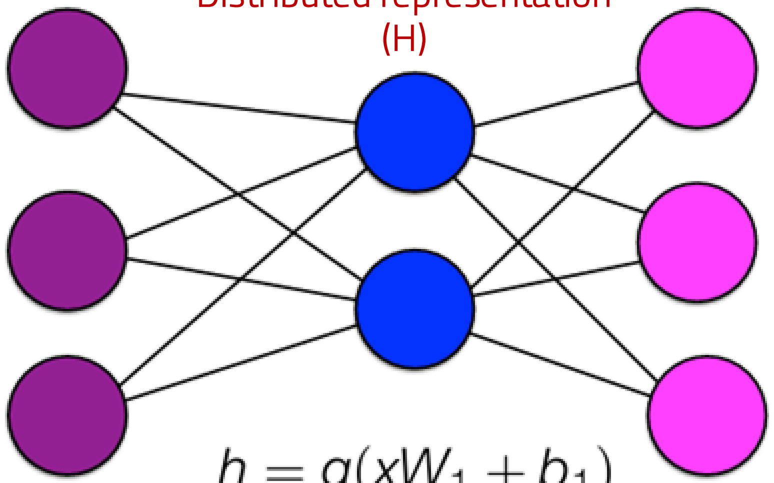
$W_1 \in \mathbb{R}^{kV \times H}$ $\quad$ $W_2 \in \mathbb{R}^{H \times V}$

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

One-hot encoding
(|x| = V)

Output space: |y| = V

Distributed representation
(H)



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

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

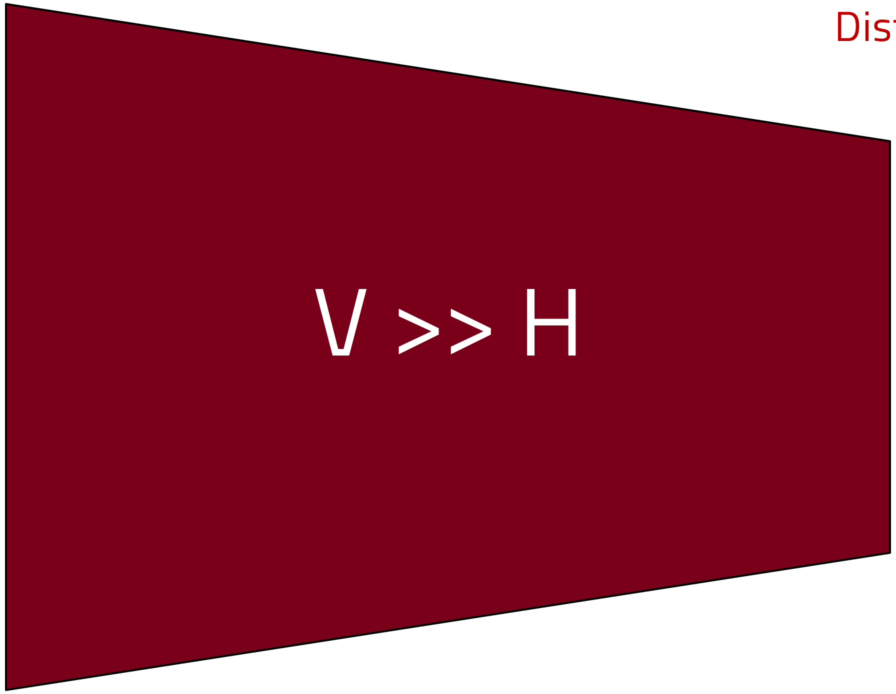
$$\hat{y} = \text{softmax}(hW_2 + b_2)$$

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.