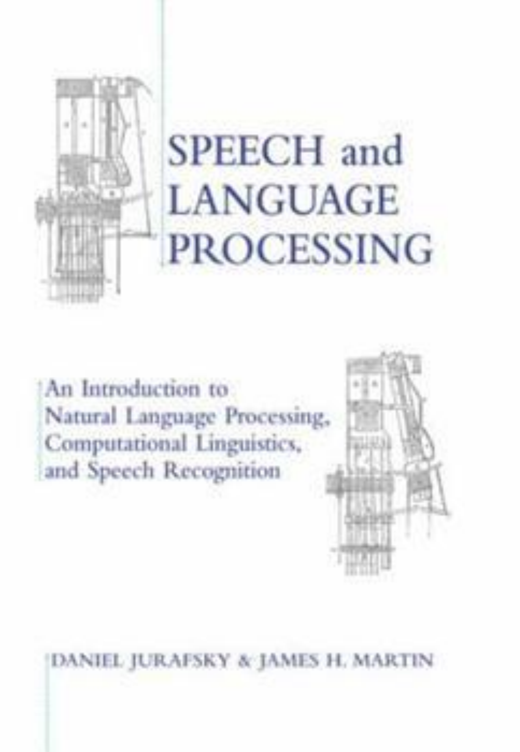
Memento, Christopher Nolan
Guy Pearce, Carrie-Anne Moss, Joe Pantoliano
September 2000
Budget $4.5 m
Box office $40 m

n-Gram Language Models



Language Modeling

CH03

Language Model

A model that can communicate

A model that generates a meaningful stream of linguistic elements

(words + inflection rules, punctuations, fillers, ...)

Language Model

How's everything? [ ]

1. Awesome. I am enjoying the sunny day in a restaurant.
2. I cannot tell you how everything is doing.
3. Mind your own business! Ha ha ha ...
4. Good. Thanks for asking

Language Model

How's everything? [a friend asks.]

1. Awesome. I am enjoying the sunny day in a restaurant.
2. I cannot tell you how everything is doing.
3. Mind your own business! Ha ha ha ...
4. Good. Thanks for asking

Language Model

How’s everything? [a close friend asks.]

1. Awesome. I am enjoying the sunny day in a restaurant.
2. I cannot tell you how everything is doing.
3. Mind your own business! Ha ha ha …
4. Good. Thanks for asking

Language Model

How's everything? [Windsor ...]
[Sitting in a restaurant ...]
[Weather is sunny ...]
[Receive a call from a close friend ...]

Language Model

$\text{Context}_i \rightarrow \text{Context}_{i+1}$


THE SOURCE
MEMORY IS TREACHERY
him

n-Gram Language Model

n-Gram Language Model

Gram = Token = Linguistic Element

Including boundaries like <w>,</w>,<s>,</s>,

[Windsor …]

[Sitting in a restaurant …]

[Weather is sunny …]

[Receive a call from a close friend …]

[<s>][How][’][s] [everything][?][</s>]

n-Gram Language Model

$$w_1 \ldots w_{n-2}\, w_{n-1} \rightarrow w_n$$

V: Vocabulary Set

$w_i \in V$, a word in V

stream of *n* grams (ordered)

n-Gram Language Model

$w_1 \dots w_{n-2}\, w_{n-1} \rightarrow w_n$ 1-gram = unigram

$w_1 \dots w_{n-2}\, w_{n-1} \rightarrow w_n$ 2-gram = bigram

$w_1 \dots w_{n-2}\, w_{n-1} \rightarrow w_n$ 3-gram = trigram

$w_1 \dots w_{n-2}\, w_{n-1} \rightarrow w_n$ n-gram

n-Gram Language Model

Context Window of Size n

Recent Past of Size n-1 → Future of Size 1

$$w_{i+1} \cdots w_{i+n-2}\; w_{i+n-1} \longrightarrow w_{i+n}$$

n-Gram Language Model

$\longrightarrow w_{i+1}$	1-gram = unigram
$w_{i+1} \longrightarrow w_{i+2}$	2-gram = bigram
$w_{i+1}\ w_{i+2} \longrightarrow w_{i+3}$	3-gram = trigram
$w_{i+1} \dots\ w_{i+n-2}\ w_{i+n-1} \longrightarrow w_{i+n}$	n-gram

n-Gram Language Model

Context Window of Size *n*

Recent Past of Size *n-1* → Future of Size 1

$$w_{i+1}^{i+n-1} \longrightarrow w_{i+n}$$

1-Gram Language Modeling

aka unigram

[Windsor ...]
[Sitting in a restaurant ...]
[Weather is sunny ...]
[Receive a call from a close friend ...]
[<s>][How]['][s] [everything][?][</s>]

1. → Random word, e.g., [nlp]
2. → Most frequent word, e.g., [the]
3. → Least frequent word, e.g., [precipitation]

2-Gram Language Modeling

aka bigram

[Windsor …]
[Sitting in a restaurant …]
[Weather is sunny …]
[Receive a call from a close friend …]
[<s>][How]['][s] [everything][?][</s>]

1. → Random word, e.g., [nlp]
2. → Most frequent word that starts a sentence, e.g., [I]
3. → Least frequent word, e.g., [precipitation]
4. → Grammatically, we start with a subject → most frequent subject

3-Gram Language Modeling

aka trigram

[Windsor …]
[Sitting in a restaurant …]
[Weather is sunny …]
[Receive a call from a close friend …]
[<s>][How]['][s] [everything][?][</s>]

1. → Random word, e.g., [nlp]
2. → Most frequent word that starts a reply, e.g., [Yes]
3. → Least frequent word, e.g., [precipitation]
4. → Grammatically, we start with a subject → most frequent subject

4-Gram Language Modeling

aka trigram

[Windsor ...]
[Sitting in a restaurant ...]
[Weather is sunny ...]
[Receive a call from a close friend ...]
[<s>][How]['][s] [everything][?][</s>]

1. → Random word, e.g., [nlp]
2. → Most frequent word that starts a reply, e.g., [ ]
3. → Least frequent word, e.g., [precipitation]
4. → Grammatically, we should start with [it]


JOHN G. RAPED AND

Frequentist Probability

as opposed to Bayesian Probability

Frequentist probability or frequentism is an interpretation of probability that defines an event's probability as the limit of its relative frequency in many trials - Wikipedia

n-Gram Language Modeling

Recent Past of Size *n*-1 → Future of Size 1 → Most Frequent Future Given the Past

$w_{i+1} \ldots w_{i+n-2}\ w_{i+n-1} \longrightarrow w_{i+n}$: Most Frequent given the past context

n-Gram Language Modeling

Recent Past of Size *n-1* → Future of Size 1 → Most Frequent Future Given the Past

$$w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n} = \mathrm{Max}\ P(w \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) \text{ in all } w \in V$$

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{\#(A,B)}{\#(B)}$$

$$w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n} = \text{Max } P(w \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) \text{ in all } w \in V$$

$$P(w \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

$$= \frac{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

$$P(w \mid w_{i+1}^{i+n-1}) = \frac{P(w_{i+1}^{i+n-1}\, w)}{P(w_{i+1}^{i+n-1})} = \frac{\#(w_{i+1}^{i+n-1}\, w)}{\#(w_{i+1}^{i+n-1})}$$

Trigram LM

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

$$P(w \mid [\text{Mr.}][\text{and}]) = \frac{P([\text{Mr.}][\text{and}]w)}{P([\text{Mr.}][\text{and}])} = \frac{\#([\text{Mr.}][\text{and}]w)}{\#([\text{Mr.}][\text{and}])} \quad \forall w \in V$$

[(1.0, 'Mrs.'), (0.0, 'zone'), (0.0, 'zombies'), (0.0, 'zinc'), (0.0, 'zeroed'), (0.0, 'zeal'), (0.0, 'youths'), (0.0, 'youthful'), ...]

Bigram LM

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

What word start sentences most often?

$$P(w \mid [.]) = \frac{P([.]w)}{P([.])} = \frac{\#([.]w)}{\#([.])} \quad \forall w \in V$$

[(0.1635, 'The'), (0.0588, '``'), (0.0429, 'He'), (0.0265, 'In'), (0.0258, 'A'), (0.0248, 'But'), (0.0245, 'It'), ..., (0.0136, 'This')]

Bigram LM

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

What word start sentences most often?

$$P(w \mid [?]) = \frac{P([?]w)}{P([?])} = \frac{\#([?]w)}{\#([?])} \qquad \forall w \in V$$

[(0.5, '?'), (0.06, '``'), (0.06, 'The'), (0.04, 'Asked'), (0.03, 'He'), (0.02, 'I'), (0.02, 'A'), (0.02, ')'), (0.01, 'Why'), ...]

Bigram LM

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

What word start sentences most often?

Impossible → 0

$$P(w \mid [.]) + P(w \mid [?]) - P(w \mid [.] \text{ and } [?]) \qquad \forall w \in V$$

$$P(A \cup B) = P(A) + P(B) - P(AB)$$

[(0.5, '?'), (0.23, 'The'), (0.12, '``'), (0.07, 'He'), (0.04, 'A'), (0.04, 'Asked'), (0.03, 'In'), (0.02, 'I'), (0.02, ')'), (0.02, 'But')]

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
	–Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
	–What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done.
	–This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in;
	–It cannot be but so.

Figure 3.3 Eight sentences randomly generated from four n-grams computed from Shakespeare’s works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Figure 3.4 Three sentences randomly generated from three n-gram models computed from 40 million words of the *Wall Street Journal*, lower-casing all characters and treating punctuation as words. Output was then hand-corrected for capitalization to improve readability.

“The whole problem with the world is that fools and fanatics are always so certain of themselves, and wiser people so full of doubts.”
— Bertrand Russell

Chain Rule of Probability

$$P(x_1\, x_2 \ldots x_n) = P(x_1)\, P(x_2 \mid x_1)\, P(x_3 \mid x_1 x_2) \ldots P(x_n \mid x_1 x_2 x_3 \ldots x_{n-1})$$

$$= \prod_{k=1}^{n} P(x_k \mid x_1 \ldots x_{k-1})$$

$$= \prod_{k=1}^{n} P(x_k \mid x_1^{k-1})$$

Chain Rule of Probability

$$P(w_1\, w_2 \dots w_n) = P(w_1)\, P(w_2 \mid w_1)\, P(w_3 \mid w_1 w_2) \dots P(w_n \mid w_1 w_2 w_3 \dots w_{n-1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1 \dots w_{k-1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Approximation to Chain Rule

Generalizability

Language is creative! A particular context might have never occurred before!

Approximation to Chain Rule

Efficiency

probability of a word given entire history, approximate the history by just the last few words

Unigram Approx.

Bag-of-Word (BoW). Why?

$$
\begin{aligned}
P(w_1\, w_2 \ldots\, w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \ldots P(w_n|w_1 w_2 w_3 \ldots w_{n-1}) \\
&= P(w_1)P(w_2|\quad)P(w_3|\qquad) \ldots P(w_n|\qquad\qquad) \\
&= P(w_1)P(w_2)P(w_3) \ldots P(w_n)
\end{aligned}
$$

Bigram Approx.

Markovian: probability of a variable depends only on the previous variable

$$P(w_1\, w_2 \ldots\, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|w_1w_2w_3 \ldots w_{n-1})$$
$$= P(w_1)P(w_2|w_1)P(w_3|\quad w_2) \ldots P(w_n|\qquad\qquad w_{n-1})$$

Trigram Approx.

$$P(w_1\ w_2 \ldots\ w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|w_1w_2w_3 \ldots w_{n-1})$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n| \qquad w_{n-2}\, w_{n-1})$$

n-gram Approx.

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

P([Mr.][and][Mrs.]) = P([Mr.])P([and]|[Mr.])P([Mrs.]|[Mr.][and])
= 0.00045851027827207127

= Bigram Approx. P([Mr.])P([and]|[Mr.])P([Mrs.]|~~[Mr.]~~[and])
≅ P([Mr.])P([and]|[Mr.])P([Mrs.]|[and])
≅ 0.000014208331509791766

= Unigram Approx. P([Mr.])P([and]|~~[Mr.]~~)P([Mrs.]|~~[Mr.][and]~~)
≅ P([Mr.])P([and])P([Mrs.])
≅ 0.00000009078228423943108

n-gram Approx.

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

P([Mr.][and][I]) = P([Mr.])P([and]|[Mr.])P([I]|[Mr.][and])
= 0.0

= Bigram Approx. P([Mr.])P([and]|[Mr.]) P([I]|~~[Mr.]~~[and])
≅ P([Mr.])P([and]|[Mr.])P([I]|[and])
≅ 0.00000175171210394693

= Unigram Approx. P([Mr.])P([and]|[Mr.])P([I]|~~[Mr.][and]~~)
≅ P([Mr.])P([and])P([I])
≅ 0.00000006422936315754214

Approx. n-gram Language Modeling

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',', '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

Make it worse!	Gives chance to new combination
P([Mr.][and][Mrs.])	P([Mr.][and][I])
0.00045851027827207127	0.0
1.4208331509791766e-05	1.75171210394693e-06
9.078228423943108e-08	6.422936315754214e-08

Lec03.ipynb

File Edit View Insert Runtime Tools Help Last edited on 21 Jan 2021

\+ Code + Text

n-gram Language Model

[ ]

```
!pip install --upgrade nltk
import nltk
nltk.download('punkt')
from nltk.util import ngrams
from collections import Counter
from nltk.corpus import brown, movie_reviews
nltk.download('brown')
nltk.download('movie_reviews')

#print(brown.categories())
#print(len(brown.words(categories='news')))
#print(brown.sents(categories=['news'])[:5])
#print(len(brown.sents(categories=['news', 'editorial', 'reviews'])))

tokens = brown.words(categories=['news'])
vocabularly = set(tokens)
n = 1
unigrams = ngrams(tokens, n)
unigrams_freq = Counter(unigrams)
print(unigrams_freq.most_common()[:10])

n = 2
bigrams = ngrams(tokens, n)
bigrams_freq = Counter(bigrams)
print(bigrams_freq.most_common()[:10])

n = 3
trigrams = ngrams(tokens, n)
trigrams_freq = Counter(trigrams)
print(trigrams_freq.most_common()[:10])


p_mrs_given_mr_and = trigrams_freq[('Mr.', 'and', 'Mrs.')] / bigrams_freq[('Mr.', 'and')]
print(p_mrs_given_mr_and)
p_mis_given_mr_and = trigrams_freq[('Mr.', 'and', 'Mis.')] / bigrams_freq[('Mr.', 'and')]
print(p_mis_given_mr_and)
```

n-Gram *Approximation* to Chain Rule
n-Gram LM

$$w_{i+1} \ldots w_{i+n\text{-}2}\, w_{i+n\text{-}1} \rightarrow w_{i+n}$$

$$P(w_{i+n} \mid w_{i+1} \ldots w_{i+n\text{-}2}\, w_{i+n\text{-}1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w_{i+n})}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

1-Gram *Approximation* to Chain Rule
1-Gram LM

$$\emptyset \rightarrow w_{i+n}$$

$$P(w_{i+n} \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w_{i+n})}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})} \cong P(w_{i+n})$$

2-Gram *Approximation* to Chain Rule
2-Gram LM

$$w_{i+n-1} \rightarrow w_{i+n}$$

$$P(w_{i+n} \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w_{i+n})}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})} \cong P(w_{i+n} | w_{i+n-1})$$

Products of Probabilities

Multiplying multiple numbers in [0, 1] results in very small number!

Products of Probabilities

Multiplying multiple numbers in [0, 1] results in very small number!

Do we need the actual probability value?

Products of Probabilities

Multiplying multiple numbers in [0, 1] results in very small number!

Do we need the actual probability value?

No! We need order of values. We want the word with max value.

Log of Probabilities

$$P(x_1) \times P(x_2) \times \ldots \times P(x_n) = \exp\left(\log P(x_1) + \log P(x_2) + \ldots + \log P(x_n)\right)$$

Log of Probabilities

$$P(x_1) \times P(x_2) \times \ldots \times P(x_n) \;\propto\; \log P(x_1) + \log P(x_2) + \ldots + \log P(x_n)$$

left and right sides have same order!

Chain Rule of Probability

$$P(w_1\, w_2 \ldots\, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|w_1w_2w_3 \ldots w_{n\text{-}1})$$

$$= \prod_{k=1}^{n} P(w_k|w_1 \ldots w_{k-1}) \rightarrow \sum_{k=1}^{n} \log P(wk|w_1 \ldots w_{k-1})$$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \rightarrow \sum_{k=1}^{n} \log P(wk|w_1^{k-1})$$

Meta’s AI chief: Three major challenges of artificial intelligence (Jan 29 2022)
https://mixed-news.com/en/metas-ai-chief-three-major-challenges-of-artificial-intelligence/

n-Gram Language Modeling

Recent Past of Size n-1 → Future of Size 1

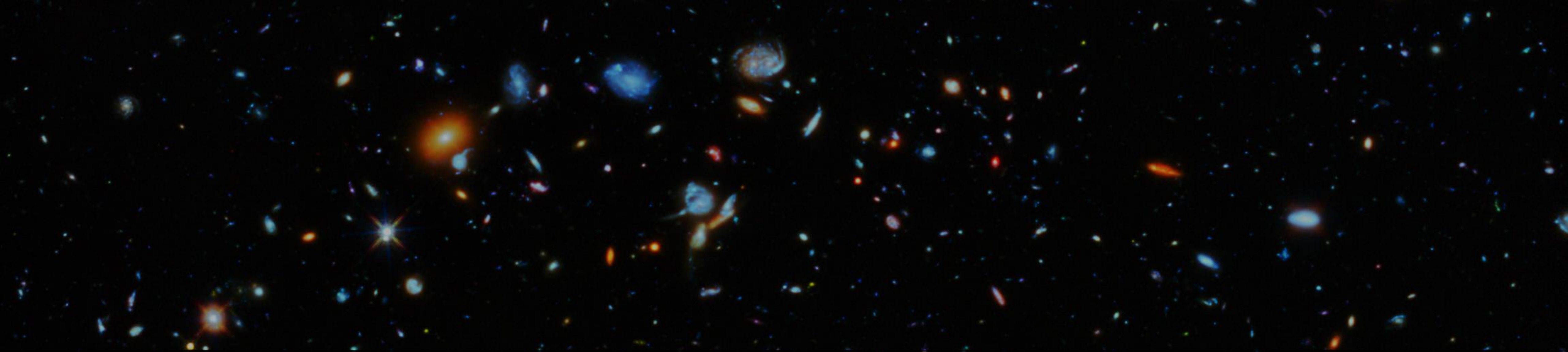
$$w_{i+1} \cdots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n}$$

Who tells us what w_{i+n} is? God, Oracle, ...

n-Gram Language Modeling

Recent Past of Size *n-1* → Future of Size 1 → Most Frequent Future Given the Past

$$w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n} = \mathrm{Max}\ P(w \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) \text{ in all } w \in V$$



n-Gram Language Modeling

Recent Past of Size *n-1* → Future of Size 1

$$w_{i+1} \cdots w_{i+n-2}\ w_{i+n-1} \longrightarrow w_{i+n}$$

Who tells us what w_{i+n} is? Data!

Self-supervised

Self-supervised learning is the key to AI understanding the world

Yann LeCun: Dark Matter of Intelligence and Self-Supervised Learning | Lex Fridman Podcast #258

https://www.youtube.com/watch?v=SGzMElJ11Cc

Tenet, Christopher Nolan,
Budget $2
Box office $363 m

n-Gram Language Modeling

Recent Past → Current ← Recent Future

$$w_{i+1} \cdots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n} \longleftarrow w_{i+n+1}\, w_{i+n+2} \cdots w_{i+n+j}$$

Following, Christopher Nolan (1998)
Budget $6,000
Box office $48,482

Evaluating Language Models

Evaluating Language Models

Unigram vs. Bigram vs. Trigram vs. *n*-gram
w/ Approx. vs. w/o Approx.

Evaluating Language Models

Higher *n* in n-gram, the better?
More history, the better prediction of future?

Evaluating Language Models

Qualitative → Let’s Communicate → Generate

Evaluating Language Models

Qualitative → Let's Communicate → Generate

```
#unigram
stream = []
while (w != '</s>'):
    w = unigrams_freq.select()
    stream.append(w)
```

Evaluating Language Models

Qualitative → Let's Communicate → Generate

```
#unigram
stream = []
while (w != '</s>'):
    w = bigrams_freq[stream[-1]].select()
    stream.append(w)
```

Evaluating Language Models

Qualitative → Let’s Communicate → Generate

```
#unigram
stream = []
while (w != '</s>'):
    w = trigrams_freq[stream[-2:]].select()
    stream.append(w)
```

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; –It cannot be but so.

Figure 3.3 Eight sentences randomly generated from four n-grams computed from Shakespeare's works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Figure 3.4 Three sentences randomly generated from three n-gram models computed from 40 million words of the *Wall Street Journal*, lower-casing all characters and treating punctuation as words. Output was then hand-corrected for capitalization to improve readability.

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

Figure 3.3 Eight sentences randomly generated from four n-grams computed from Shakespeare’s works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.

1 gram	Months the my and issue of year foreign new exchange’s september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

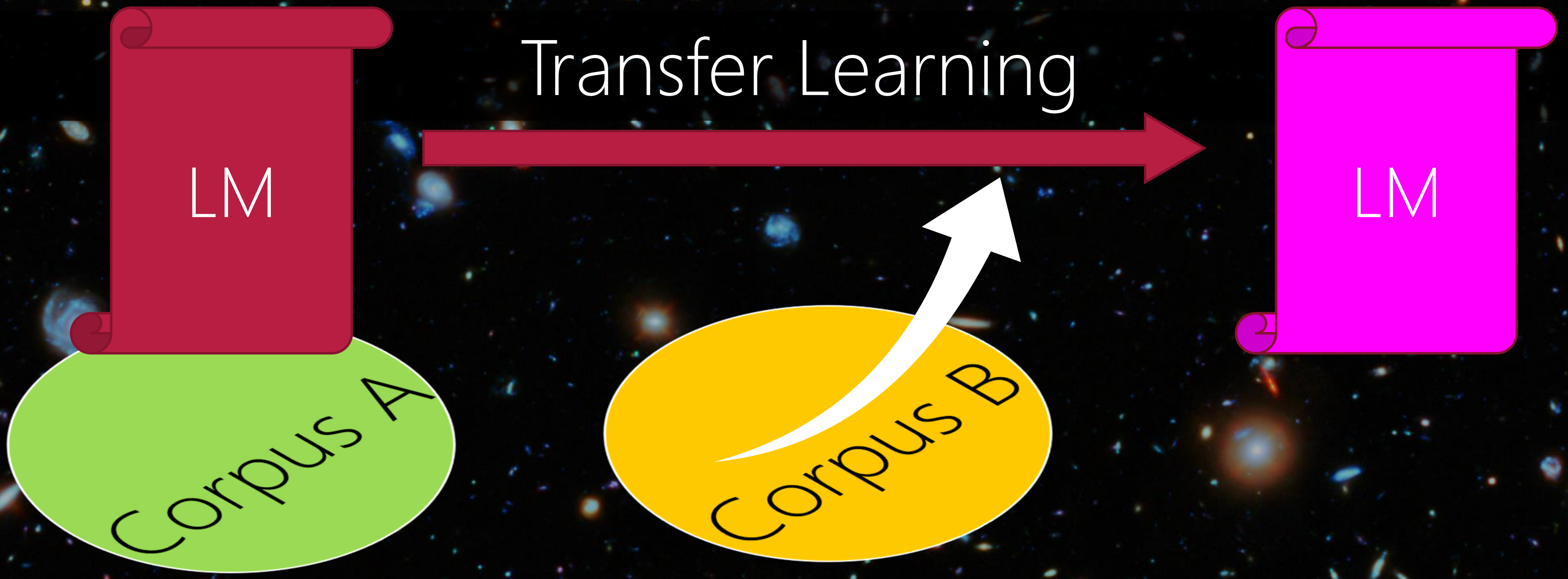
Figure 3.4 Three sentences randomly generated from three n-gram models computed from 40 million words of the *Wall Street Journal*, lower-casing all characters and treating punctua-

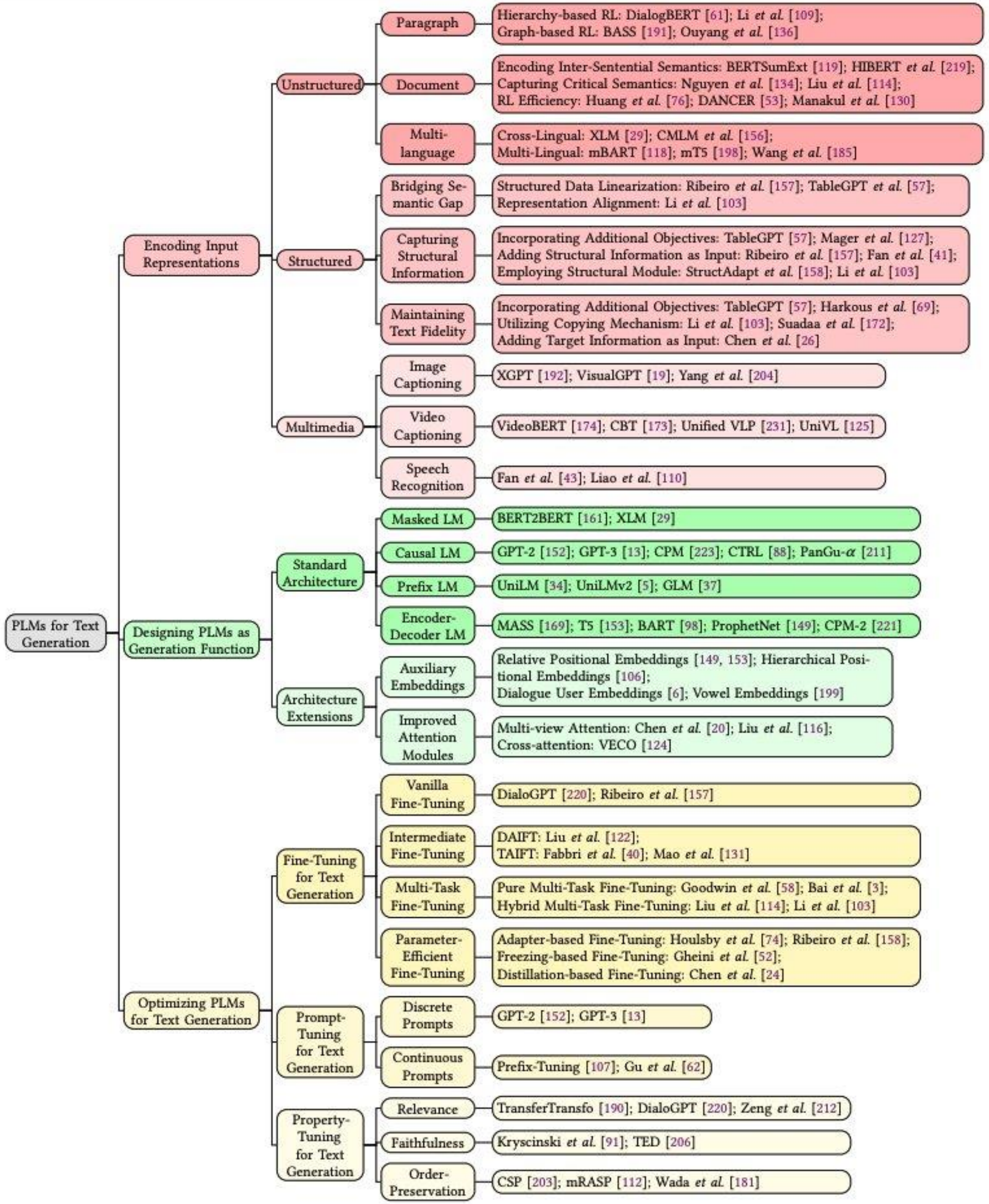
Cross Evaluating Language Models

Biased toward the corpus! Dialect, Genre, …

Better LM is the one that can generalize!

Pre-trained Language Models





elvis
@omarsar0

https://github.com/omarsar

A Survey of Pretrained Language Models Based Text Generation

Nice survey paper on recent advances in pretrained language models for text generation.
arxiv.org/abs/2201.05273

9:25 AM · Jan 17, 2022 · Twitter Web App

Evaluating Language Models

Quantitative → Likelihood