N-GRAM LANGUAGE MODELS

Language Modeling

Language Modeling

Building a model that can generate an accurate stream of tokens (words + inflection rules, punctuations, fillers, …)

Language Modeling

Context:

1. “How’s everything?”, somebody asks?

Language Modeling

Possible stream of tokens:

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 Modeling

Context:

1. “How’s everything?”, a friend asks?

Language Modeling

Possible stream of tokens:

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 Modeling

Context:

1. “How’s everything?”, a close friend asks?

Language Modeling

Possible stream of tokens:

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. Mind your own business!

Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.

Language Modeling

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

n-gram Language Modeling

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

V: Vocabulary Set

$w_i \in V$, a word in V

a stream of n tokens (ordered)

n-gram Language Modeling

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

V: Vocabulary Set

$w_i \in V$, a word in V

a stream of n tokens (ordered)

1-gram Language Modeling

unigram LM

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

$\emptyset \rightarrow w_{i+1}$

V: Vocabulary Set

$w_i \in V$, a word in V

~~a stream of~~ 1 token ~~(ordered)~~

2-gram Language Modeling

bigram LM

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

$w_{i+1} \rightarrow w_{i+2}$

V: Vocabulary Set

$w_i \in V$, a word in V

a stream of 2 tokens (ordered)

3-gram Language Modeling

trigram LM

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

$w_{i+1}\ w_{i+2} \longrightarrow w_{i+3}$

V: Vocabulary Set

$w_i \in V$, a word in V

a stream of 3 tokens (ordered)

n-gram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.

n-gram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.

n-gram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.
5. [how][is][everything][?][</s>]

unigram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.
5. [how][is][everything][?][</s>] → Randomly selected word from V, e.g., [nlp]
6. [how][is][everything][?][</s>] → Most frequent word in V, e.g., [the]
7. [how][is][everything][?][</s>] → Least frequent word in V, e.g., [precipitation]

bigram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.
5. [how][is][everything][?][</s>]→ A word in V that comes after [</s>] most frequently, i.e., start a sentence!

→ [it]

→ [i]

→ [you]

trigram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.
5. [how][is][everything][?][</s>]→ A word in V that comes after [?][</s>] most frequently, i.e., start an answer!

→ [you]

4-gram Language Modeling

Context:

1. Sitting in a restaurant in the middle of Windsor
2. The weather is sunny
3. You receive a call from a close friend.
4. “How’s everything?”, your close friend asks.
5. [how][is][everything][?][</s>]→ A word in V that comes after [everything][?][</s>] most frequently

→ [it]

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

$$w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n}$$
$$P(w_{i+1} \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1})$$

w_{i+n} could be any word in V but we pick most frequent one!

n-gram Language Modeling

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

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

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

n-gram Language Modeling

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

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

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

Trigram Language Modeling

Corpus: Brown University

Sample Sentences:

['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', "''", '.']

$$P(w|\ [Mr.][and]) = \frac{P([Mr.][and]w)}{P([Mr.][and])} = \frac{\#([Mr.][and]w)}{\#([Mr.][and])}$$

$\forall w \in V$

[(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 Language Modeling

Corpus: Brown University

Sample Sentences:

['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?

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

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

Bigram Language Modeling

Corpus: Brown University

Sample Sentences:

['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?

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

b) $P(w|[?]) = \frac{P([?]w)}{P([?])} = \frac{\#([?]w)}{\#([?])} \quad \forall w \in V$

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

Bigram Language Modeling

Corpus: Brown University

Sample Sentences:

['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?

$P(w|\ [.]) + P(w|\ [?]) - P(w|\ [\text{both '.' and '?'}])\ \forall w \in V$

*From Probability: $P(A \cup B) = P(A) + P(B) - P(AB)$

Bigram Language Modeling

Corpus: Brown University

Sample Sentences:

['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?

$P(w|\ [.]) + P(w|\ [?]) - 0\ \forall w \in V$

[(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')]

*From Probability: $P(A \cup B) = P(A) + P(B) - P(AB)$

Chain Rule of Probability

$$P(X_1\, X_2 \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1X_2) \ldots P(X_n|X_1X_2X_3 \ldots X_{n-1})$$
$$= \prod_{k=1}^{n} P(Xk|X_1 \ldots X_{k-1})$$
$$= \prod_{k=1}^{n} P(Xk|X_1^{k-1})$$

Chain Rule of Probability

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

Approximation to Chain Rule

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

- **Efficiency**: Instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

Approximation to Chain Rule

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

- **Generalizability**: Language is creative and any particular context might have never occurred before! We can't just estimate by counting the number of times every word occurs following every long string.

Unigram Approximation

Bag-of-Word (BoW)

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

Bigram Approximation

Markovian

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

The assumption that the probability of a variable depends only on the previous variable is called Markov assumption.

Trigram Approximation

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

Approx. n-gram Language Modeling

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

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

2-gram LM: $P(w_{i+n} \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) \cong P(w_{i+n} \mid w_{i+n-1})$

3-gram LM: $P(w_{i+n} \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) \cong P(w_{i+n} \mid w_{i+n-2}\, w_{i+n-1})$

Approx. n-gram Language Modeling

Corpus: Brown University

Sample Sentences:

['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', "''", '.']

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

$P([Mr.][and][Mrs.]) = P([Mr.])P([and]|[Mr.])P([Mrs.]|[Mr.][and])$ Bigram Approx.

$\cong P([Mr.])P([and]|[Mr.])P([Mrs.]|[and])$

$\cong 0.00001420833150979176$

$P([Mr.][and][Mrs.]) = P([Mr.])P([and]|[Mr.])P([Mrs.]|[Mr.][and])$ Unigram Approx.

$\cong P([Mr.])P([and])P([Mrs.])$

$\cong 0.000000090782284239431 08$

Approx. n-gram Language Modeling

Corpus: Brown University

Sample Sentences:

['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', "''", '.']

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

$P([Mr.][and][I]) = P([Mr.])P([and]|[Mr.])P([Mrs.]|[Mr.][and])$ Bigram Approx.

$\cong P([Mr.])P([and]|[Mr.])P([Mrs.]|[and])$

$\cong 0.00000175171210394693$

$P([Mr.][and][I]) = P([Mr.])P([and]|[Mr.])P([Mrs.]|[Mr.][and])$ Unigram Approx.

$\cong P([Mr.])P([and])P([Mrs.])$

$\cong 0.00000006422936315754214$

Approx. n-gram Language Modeling

Corpus: Brown University

Sample Sentences:

['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.][and][I])
0.0004585102782720712 7	0.0
1.4208331509791766e-05	1.751712103946 93e-06
9.078228423943108e-08	6.422936315754214e-08

Log Probability

$$P(X_1) \times P(X_2) \times \ldots \times P(X_n) = \exp(\log P(X_1) + \log P(X_2) + \ldots + \log P(X_n))$$

Log Probability

$$P(X_1) \times P(X_2) \times \ldots \times P(X_n) = \exp(\log P(X_1) + \log P(X_2) + \ldots + \log P(X_n))$$

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

Log Probability

$$P(X_1) \times P(X_2) \times \ldots \times P(X_n) = \exp(\log P(X_1) + \log P(X_2) + \ldots + \log P(X_n))$$

- Multiplying multiple numbers in [0, 1] results in very small number!
- We don't care about the actual probability value but the relative order. Our task is often the selection of the most probable item.

Log Probability

$$P(X_1) \times P(X_2) \times \ldots \times P(X_n) = \exp(\log P(X_1) + \log P(X_2) + \ldots + \log P(X_n))$$

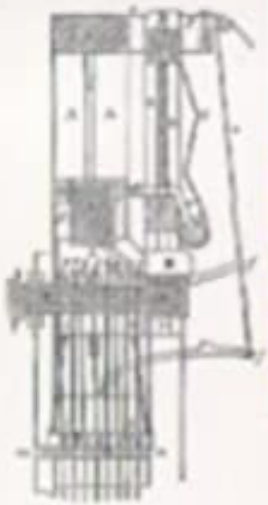
- Multiplying multiple numbers in [0, 1] results in very small number!
- We don't care about the actual probability value but the relative order. Our task is often the selection of the most probable item.
- Optimizing the left and right sides have the same effect.

Chain Rule of Probability

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

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

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



SPEECH and LANGUAGE PROCESSING

An Introduction to
Natural Language Processing,
Computational Linguistics,
and Speech Recognition

DANIEL JURAFSKY & JAMES H. MARTIN