

Task of assigning probability to a sentence or a phrase

$$P(S) = P(W_1, W_2, W_3, ..., W_n)$$

• It can also be used to compute the probability of upcoming words

$$P(w_5 | w_1, w_2, w_3, w_4)$$



$$P(A|B) = P(A,B) / P(B)$$





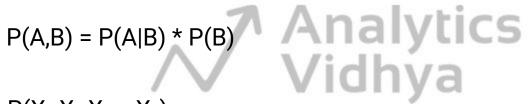
$$P(A|B) = P(A,B) / P(B)$$



$$P(A|B) = P(A,B) / P(B)$$

$$P(A,B) = P(A|B) * P(B)$$

$$P(X_1, X_2, X_3, ..., X_n) =$$

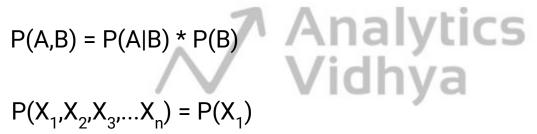




$$P(A|B) = P(A,B) / P(B)$$

$$P(A,B) = P(A|B) * P(B)$$

$$P(X_1, X_2, X_3, ... X_n) = P(X_1)$$





$$P(A|B) = P(A,B) / P(B)$$

 $P(A,B) = P(A|B) * P(B)$
 $P(X_1,X_2,X_3,...X_n) = P(X_1) * P(X_2|X_1)$



$$P(A|B) = P(A,B) / P(B)$$

$$P(X_1, X_2, X_3, ...X_n) = P(X_1) * P(X_2|X_1) * P(X_3|X_1, X_2)$$



$$P(A|B) = P(A,B) / P(B)$$

$$P(X_1, X_2, X_3, ..., X_n) = P(X_1) * P(X_2|X_1) * P(X_3|X_1, X_2) * ... * P(X_n|X_1, X_2, ..., X_{n-1})$$



















can you come here

P(can,you,come,here) = P(can) * P(you|can) * P(come|can,you) * P(here|can,you,come)

P(here|can,you,come) = ?



can you come here

P(can,you,come,here) = P(can) * P(you|can) * P(come|can,you) * P(here|can,you,come)

P(here|can,you,come) = Count of ("can you come here")



can you come here

P(can,you,come,here) = P(can) * P(you|can) * P(come|can,you) * P(here|can,you,come)



What if there is no sentence "can you come here"

P(can,you,come,here) = P(can) * P(you|can) * P(come|can,you) * P(here|can,you,come)



- What if there is no sentence "can you come here"
 - Count of ("can you come here) = 0



- What if there is no sentence "can you come here"
 - Count of ("can you come here) = 0
 - then P(here | can, you, come) = 0

P(can,you,come,here) = P(can) * P(you|can) * P(come|can,you) * P(here|can,you,come)



P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)Count of (w1, w2, w3, w4)





$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

• Markov Assumption: P(w5 | w1, w2, w3, w4) ≈ P(w5 | w3, w4)



$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

- Markov Assumption: $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$
- $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$



$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

- Markov Assumption: $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$
- $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$
- P(here | can, you, come) ≈ P(here | come)



$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

- Markov Assumption: $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$
- $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$
- P(here | can, you, come) ≈ P(here | come) = Count of (come here)
 Count of (come)



$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

- Markov Assumption: $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$
- $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$
- P(here | can, you, come) ≈ P(here | come) = Count of (come here)
 Count of (come)
- P(here | can, you, come) ≈ P(here | you, come)



$$P(w5 | w1, w2, w3, w4) = Count of (w1, w2, w3, w4, w5)$$

Count of (w1, w2, w3, w4)

- Markov Assumption: $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$
- $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$
- P(here | can, you, come) ≈ P(here | come) = Count of (come here)
 Count of (come)
- P(here | can, you, come) ≈ P(here | you, come) = Count of (you come here)
 Count of (you come)







An N-gram is a sequence of N tokens.





can you come here

An N-gram is a sequence of N tokens.





can you come here

An N-gram is a sequence of N tokens.

Unigrams:

N = 1, ["can", "you", "come", "here"]





can you come here

An N-gram is a sequence of N tokens.

Unigrams:

N = 1, ["can", "you", "come", "here"]

 $P(w5 | w1, w2, w3, w4) \approx P(w5)$



can you come here

An N-gram is a sequence of N tokens. Analytics Vidhya

Unigrams:

N = 1, ["can", "you", "come", "here"]

Bigrams:

N = 2, ["can you", "you come", "come here"]



can you come here

An N-gram is a sequence of N tokens. Analytics Vidhya

Unigrams:

N = 1, ["can", "you", "come", "here"]

Bigrams:

N = 2, ["can you", "you come", "come here"]

 $P(w5 | w1, w2, w3, w4) \approx P(w5 | w4)$



can you come here

An N-gram is a sequence of N tokens. Analytics Vidhya

Unigrams:

N = 1, ["can", "you", "come", "here"]

Bigrams:

N = 2, ["can you", "you come", "come here"]

Trigrams:

N = 3, ["can you come", "you come here"]



can you come here

An N-gram is a sequence of N tokens.

Unigrams: N = 1, ["can", "you", "come", "here"]

Bigrams:

N = 2, ["can you", "you come", "come here"]

Trigrams:

N = 3, ["can you come", "you come here"]

 $P(w5 | w1, w2, w3, w4) \approx P(w5 | w3, w4)$





