Challenges in N-Gram Language Modeling

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”)

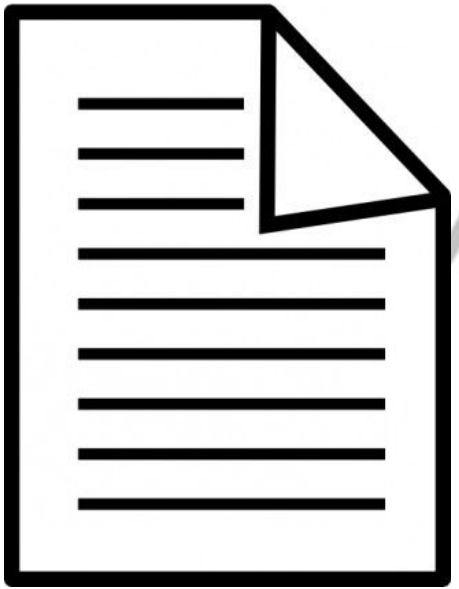
Analytics Vidhya

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”) =

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”)

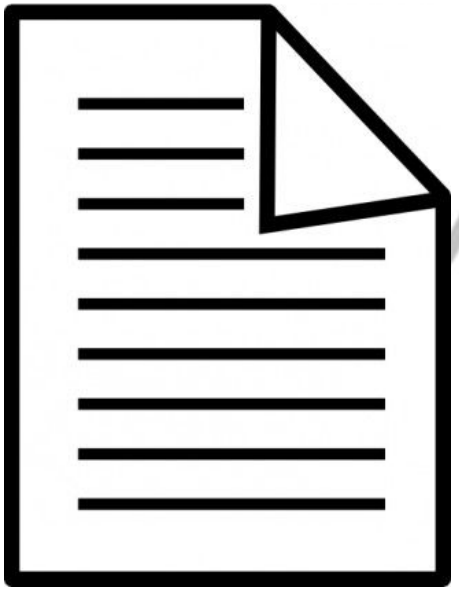
N-Gram Language Modeling - Challenges



"Jack" appears 50 times

Total words = 500

N-Gram Language Modeling - Challenges

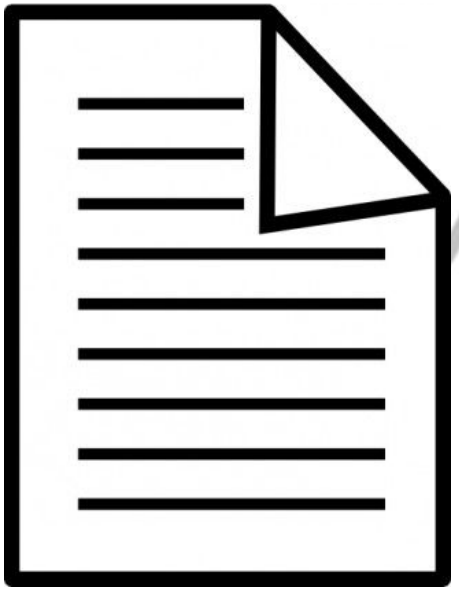


"Jack" appears 50 times

Total words = 500

P("Jack") = 50/500 = 0.1

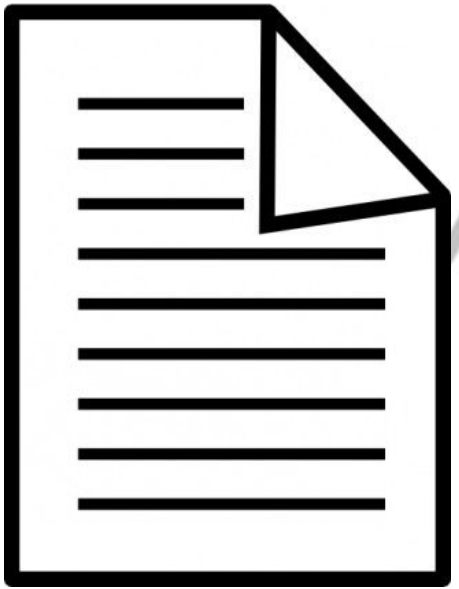
N-Gram Language Modeling - Challenges



10 sequences start with “Jack”

Total words = 500

N-Gram Language Modeling - Challenges



10 sequences start with “Jack”

Total words = 500

P(“Jack”) = 10/500 = 0.02

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”) =

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”)

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”) =

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”)
- Add a start token
  - P(“ <s> Jack built that house”)

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”
- Model output = “Jack built that house in two years…”

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”
- Model output = “Jack built that house in two years in...”

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”
- Model output = “Jack built that house in two years in India...”

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”
- Model output = “Jack built that house in two years in India is...”

N-Gram Language Modeling - Challenges

- Input = “Jack built that house”
- Model output = “Jack built that house in two years in India is...”
- Model does not know when to stop text generation

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”) =

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”)
- Add Start Token
  - P(“ <s> Jack built that house”)
- Add End Token
  - P(“ <s> Jack built that house </s>”)

N-Gram Language Modeling - Challenges

- N = 3 (Tri-gram model)
- P(“Jack built that house”) =

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”)

- Add Start Token
  - P(“ <s> Jack built that house”)
- Add End Token
  - P(“ <s> Jack built that house </s>”)

  P(“Jack”). P(“built”|”Jack”). P(“that”|”Jack built”). P(“house”|”built that”) P(“</s>”|”that house”)

N-Gram Language Modeling - Challenges

Limitations of N-Gram LM

- The higher the N, the better is the model usually. But this leads to lots of computation overhead that requires large computation power.

N-Gram Language Modeling - Challenges

Limitations of N-Gram LM

- The higher the N, the better is the model usually. But this leads to lots of computation overhead that requires large computation power.

- **Sparsity Problem**: N-grams are a sparse representation of language. It will give zero probability to all the words that are not present in the training corpus.

N-Gram Language Modeling - Challenges

- Training Set
    - ...find the differences
    - ...find the station
    - ...find the jacket

Analytics Vidhya

N-Gram Language Modeling - Challenges

- Training Set
  - ...find the differences
  - ...find the station
  - ...find the jacket
- Unseen Set
  - ...find the cat

Analytics Vidhya

N-Gram Language Modeling - Challenges

- Training Set
  - ...find the differences
  - ...find the station
  - ...find the jacket
- Unseen Set
  - ...find the cat
- P(“cat” | “find the”) = 0

Analytics Vidhya

N-Gram Language Modeling - Challenges

- **Laplace Smoothing:** Add one to the occurrence of each word

N-Gram Language Modeling - Challenges

- **Laplace Smoothing:** Add one to the occurrence of each word
- Just add one to all the counts

Analytics Vidhya

N-Gram Language Modeling - Challenges

- **Laplace Smoothing:** Add one to the occurrence of each word
- Just add one to all the counts

“Without Smoothing”

P(“cat” | “find the”)	0	0/50

N-Gram Language Modeling - Challenges

- **Laplace Smoothing:** Add one to the occurrence of each word
- Just add one to all the counts

	"Without Smoothing"		"With Smoothing"	
P("cat" | "find the")	0	0/50	1	1/51

N-Gram Language Modeling - Challenges

- **Backoff:** Switch to shorter context/history as an approximation
    - P(“offer”| ”denied the”) -> P(“offer”| ”the”) -> P(“offer”)

N-Gram Language Modeling - Challenges

- **Backoff:** Switch to shorter context/history as an approximation
  - P(“offer”| ”denied the”) -> P(“offer”| ”the”) -> P(“offer”)
- **Interpolation:** Use weighted sum of unigrams, bigrams, and trigrams
  - $P(word_n \mid word_{n-1}\ word_{n-2}) = a_1 P(word_n \mid word_{n-1}\ word_{n-2}) + a_2 P(word_n \mid word_{n-1}) + a_3 P(word_n)$
  - $\Sigma a_i = 1$

N-Gram Language Modeling - Challenges

- “There she built a ? “

N-Gram Language Modeling - Challenges

- “There she built a ? “
- “Alice went to the beach. There she built a ? ”

N-Gram Language Modeling - Challenges

- “There she built a ? “
- “Alice went to the beach. There she built a sandcastle ”

Long Term Dependency

Thank You

Analytics
Vidhya