# Language Models

## N-Gram

Dr. Uzair Ahmad

# Program

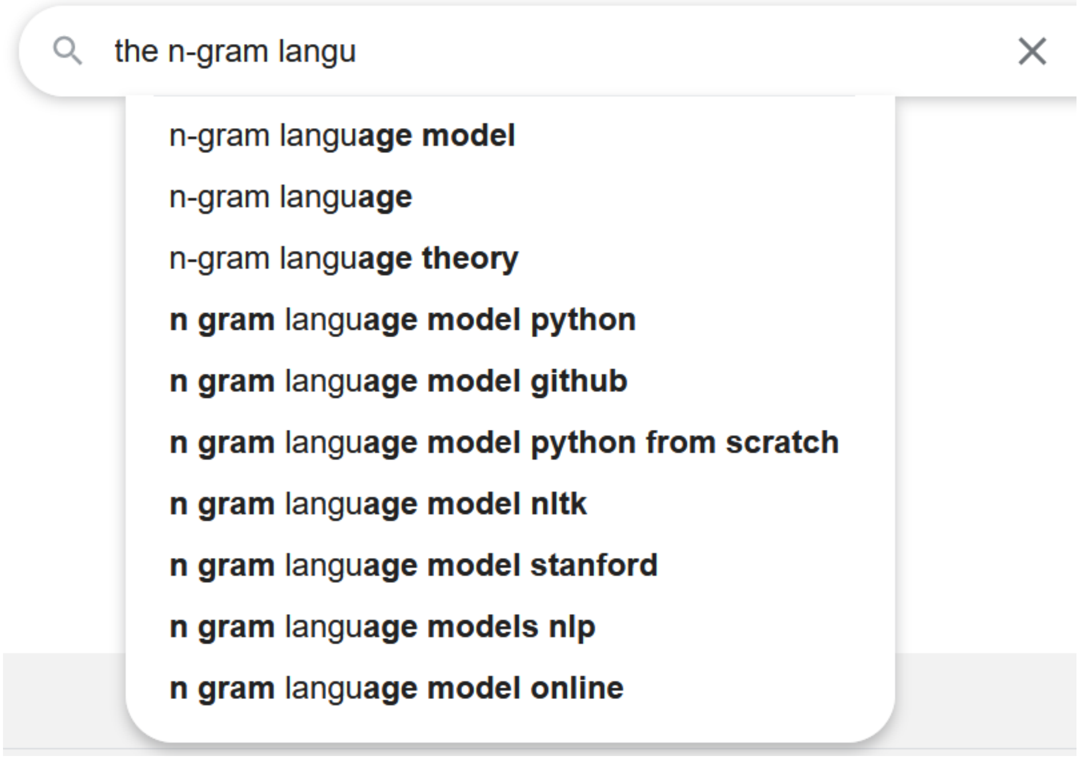
- N-Gram Language Models
  - Predict next word
  - Predict a sentence
- Evaluation of language models

# Motivation

- You are uniformly charming!" cried he, with a smile of associating and now and then I bowed and they perceived a chaise and four to wish for.

Random sentence generated from a Jane Austen trigram model

# Predicting the next word



# Probability of a sequence

- Relative frequency approach

**Unbiased estimate** $$\mathrm{p}(\textit{Computers are useless, they can only give you answers}) = \frac{\mathrm{count}(\textit{Computers are useless, they can only give you answers})}{\mathrm{count}(\text{all sentences ever spoken})}$$

Length 9

How many examples are good enough ?

- [Colorless green ideas sleep furiously](https://en.wikipedia.org/wiki/Colorless_green_ideas_sleep_furiously)

https://en.wikipedia.org/wiki/Colorless_green_ideas_sleep_furiously

# N-Gram Language Models

- Chain rule refactoring approach

**Biased estimate**

$$p(\boldsymbol{w}) = p(w_1, w_2, w_3 \dots w_M)$$
$$p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \dots p(w_M|w_{M-1}, \dots, w_1)$$

$$\mathrm{p}(\textit{useless} \mid \textit{computers are}) = \frac{\mathrm{count}(\textit{computers are useless})}{\sum_{x \in \mathcal{V}} \mathrm{count}(\textit{computers are } x)}$$
$$= \frac{\mathrm{count}(\textit{computers are useless})}{\mathrm{count}(\textit{computers are})}.$$

# N-Gram Language Models

- The chain rule of probability
  - Product of probabilities of subsequences
    - $p(\boldsymbol{w}) = p(w_1, w_2, w_3 \dots w_M)$
    - $p(\boldsymbol{w}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \dots p(w_M|w_{M-1}, \dots, w_1)$
  - Reverse order
    - $p(\boldsymbol{w}) = p(w_M)p(w_{M-1}|w_M)p(w_{M-2}|w_{M-1}, w_M) \dots p(w_1|w_2, \dots, w_M)$
    - $\prod_{m=1}^{M} p(w_M|w_{M-1}, \dots, w_1)$

# N-Gram Language Models

$$p(\boldsymbol{w}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \dots\ p(w_M|w_{M-1}, \dots, w_1)$$

$$p(w_M|w_{M-1}, \dots, w_1) \approx p(w_M|w_{M-1}, \dots, w_{M-n+1})$$

$$p(\boldsymbol{w}) = p(w_1, w_2, w_3 \dots\ w_M) \approx \prod_{m=1}^{M} p(w_m|w_{m-1}, \dots, w_{m-n+1})$$

$$\boxed{\prod_{m=1}^{M} p(w_m|w_1\!:w_{m-1}) \approx \prod_{m=1}^{M} p(w_{m-N+1}|w_{m-1})}$$

# The Markov assumption

$$\prod_{m=1}^{M} p(w_m | w_1 : w_{m-1}) \approx \prod_{m=1}^{M} p(w_{m-N+1} | w_{m-1})$$

$$p(I, like, black, coffee) \approx$$
$$p(I \mid \odot) \times$$
$$p(like \mid I) \times$$
$$p(black \mid like) \times$$
$$p(cofee \mid black) \times$$
$$p(\otimes \mid cofee)$$

How many contexts needed for N-Gram Model ?

# N-Gram Language Models

- A Bi-Gram model

$$p(w_m|w_{m-1}) = \frac{count(w_m, w_{m-1})}{\sum_{w'} count(w', w_{m-1})} = \frac{count(w_m, w_{m-1})}{count(w')}$$

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

$P(\texttt{I}|\texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam}|\texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am}|\texttt{I}) = \frac{2}{3} = .67$

$P(\texttt{</s>}|\texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam}|\texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do}|\texttt{I}) = \frac{1}{3} = .33$

# N-Gram Language Models

- An N-Gram model

$$p(w_m | w_{m-N+1} : w_{m-1}) = \frac{count(w_{m-N+1} : w_{m-1} w_m)}{count(w_{m-N+1})}$$

Relative Frequency

- The hyper-parameter N:
  - **Gorillas** always like to groom **their** friends
    - Too small → High bias:
    - Too big → Large variance

# N-Gram Language Models

- Large N but … smoothing for Unknowns

$$p_{smooth}(w_m|w_{m-1}) = \frac{count(w_m, w_{m-1}) + \alpha}{\sum_{w'} count(w', w_{m-1}) + |V|\alpha}$$

- Lidstone smoothing
  - Laplace smoothing $\alpha = 1$
  - Jeffreys-Perks law $\alpha = 0.5$

# N-Gram Language Models

- Effective counts

$$p_{smooth}(w_m|w_{m-1}) = \frac{count(w_m, w_{m-1}) + \alpha}{\sum_{w'} count(w', w_{m-1}) + |V|\alpha}$$

$$\boxed{c_i^* = (c_i + \alpha)\frac{M}{M + |V|\alpha}}$$

- Discount for each n-gram

$$\boxed{d_i = \frac{c_i^*}{c_i} = \frac{(c_i + \alpha)}{c_i}\frac{M}{M + |V|\alpha}}$$

# N-Gram Language Models

- Smoothing and Absolute Discounting

| | counts | unsmoothed probability | Lidstone smoothing, $\alpha = 0.1$ | | Discounting, $d = 0.1$ | |
|---|---|---|---|---|---|---|
| | | | effective counts | smoothed probability | effective counts | smoothed probability |
| *impropriety* | 8 | 0.4 | 7.826 | 0.391 | 7.9 | 0.395 |
| *offense* | 5 | 0.25 | 4.928 | 0.246 | 4.9 | 0.245 |
| *damage* | 4 | 0.2 | 3.961 | 0.198 | 3.9 | 0.195 |
| *deficiencies* | 2 | 0.1 | 2.029 | 0.101 | 1.9 | 0.095 |
| *outbreak* | 1 | 0.05 | 1.063 | 0.053 | 0.9 | 0.045 |
| *infirmity* | 0 | 0 | 0.097 | 0.005 | 0.25 | 0.013 |
| *cephalopods* | 0 | 0 | 0.097 | 0.005 | 0.25 | 0.013 |

# N-Gram Language Models

- Backoff

$$c^*(i,j) = c(i,j) - d$$

$$\mathrm{p}_{\mathrm{Katz}}(i \mid j) = \begin{cases} \frac{c^*(i,j)}{c(j)} & \text{if } c(i,j) > 0 \\ \alpha(j) \times \frac{\mathrm{p}_{\mathrm{unigram}}(i)}{\sum_{i':c(i',j)=0} \mathrm{p}_{\mathrm{unigram}}(i')} & \text{if } c(i,j) = 0. \end{cases}$$

# N-Gram Language Models

- Interpolation

$$\begin{aligned}
\mathrm{p}_{\text{Interpolation}}(w_m \mid w_{m-1}, w_{m-2}) = {} & \lambda_3 \mathrm{p}_3^*(w_m \mid w_{m-1}, w_{m-2}) \\
& + \lambda_2 \mathrm{p}_2^*(w_m \mid w_{m-1}) \\
& + \lambda_1 \mathrm{p}_1^*(w_m).
\end{aligned}$$

# Evaluation of N-Gram Language Models

- Extrinsic
  - Model embedded in Application
- Intrinsic
  - Train-Test Split
  - K-Fold Cross Validation

# Evaluation of N-Gram Language Models

- The Perplexity

$$PP(w) = P(w_1, w_2, w_3 \dots w_M)^{-\frac{1}{M}}$$

$$PP(w) = \sqrt[M]{\frac{1}{P(w_1, w_2, w_3 \dots w_M)}}$$

$$PP(w) = \sqrt[M]{\frac{1}{\prod_{i=1}^{M} P(w_i | w_1, w_2 \dots w_{i-1})}}$$

# Summary and Questions

- N-Gram Language Models
  - Predict next word
  - Predict a sentence
- Smoothing relative frequencies
- Evaluation of language models
  - Intrinsic: Perplexity
  - Extrinsic