# Sequence Learning Markov Chains and n-grams

### **Korbinian Riedhammer**

TECHNISCHE HOCHSCHULE NÜRNBERG GEORG SIMON OHM

### Markov Assumption Next prediction depends only on the recent (rather than entire) history

1st order Markov assumption



# Some Terminology

- Type: unique symbol or word (programming: class)
- Token: instance of a type or word occurrence (programming: object)

### Make America great again! 5 types and tokens



### 5 tyes and 8 tokens



### ccdcfe ccdcgf ccCafed bbafgf

- 8 types: Cabcdefg
- 25 tokens
- Probability of each type?
- Probability of type given current token?

$$p(w_n | w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

import pandas as pd # happy birthday to you s = 'ccdcfeccdcgfccCafedbbafgf' df = pd.DataFrame(list(s))

print(df.nunique()) print(df.value\_counts()) print(df.value\_counts() / len(df))

$$p(w_n) = \frac{C(w_n)}{\sum_j C(w_j)}$$

Table

С	0.
f	0.
d	0.
а	0.
b	0.
е	0.
g	0.
С	0.







<ccdcfeccdcgfccCafedbbafgf/>

$$p(c \mid d) = \frac{C(dc)}{\sum_{x} C(dx)} = \frac{C(dc)}{C(dx)}$$

1st order Markov Chain ~ bi-gram









# **Probability of Sequences**

- Sequence or "sentence"  $w = w_1 w_2 \cdots w_n$
- Expand via chain rule  $p(w) = p(w_1, w_2, ..., w_n) = p(w_1) \cdot p(w_2 | w_1) \cdots p(w_n | w_1 \dots w_{n-1})$
- Probability of  $w_i$  depends on all prior words (=history)
- $p(w_i | w_1 w_2 \cdots w_{n-1})$  cannot be estimated for large *i*
- Limit context by defining equivalence classes for preceding words

### **Probability of Sequences** ... of limited context

- Map longer histories to same (shortened) history  $p(\boldsymbol{w}) = p(w_1) \cdot \prod p(w_i | \phi(w_1 \cdots w_{i-1}))$ i=2
- Bi-gram *language model*  $p(\boldsymbol{w}) = p(w_1) \cdot \prod p(w_i | w_{i-1})$ i=2
- Tri-gram language model  $p(\mathbf{w}) = p(w_1) \cdot p(w_2 | w_1) \cdot \prod_{i=1}^{N} p(w_i | w_i)$ i=3

### Compute in log-domain to avoid underflows!

$$\mathcal{V}_{i-2}\mathcal{W}_{i-1}$$
)



### **N-gram models** ...or (n-1)-order Markov chains

• Uni-gram: 
$$p(w_n) = \frac{C(w_n)}{\sum_j C(w_j)}$$
  
• Bi-gram:  $p(w_n | w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$ 

• N-gram:  $p(w_n | w_{n-N+1}, ..., w_{n-1}) =$ 

$$= \frac{C(w_{n-N+1}...w_n)}{C(w_{n-N+1}...w_{n-1})}$$

# **ML Estimates of N-Gram Language Models**

- Compute relative frequencies for each 1...n-gram (formulas on prev. slide)
- Relative frequencies ~ ML estimate (for exhaustive context size)

### **Trumps Language** https://www.thetrumparchive.com

- ca. 45k tweets (excluding retweets)
- account permanently suspended Jan 8, 2021
- ca. 44k types, including
  - ca. 15k mentions
  - ca. 2k hashtags
  - ca. 9k that appear more than 5 times

	Word counts for Trump's	tweets
1	<s></s>	45
2		45
3	the	38
4	to	21
5	and	17
6	a	16
7	of	14
8	is	13
9	in	13
10	for	10
11	you	10
12	İ	9
13	on	8
14	@realdonaldtrump	8
15	will	7
16	be	7
17	great	7



# Zipf Distribution



# Natural Language "Generation" ...by randomly sampling from n-grams

- if only donald trump appearing today on crony green energy projects
- @srwmichellej @realdonaldtrump and @terrellowens sounds like it thugs
- at 9 00 with top representatives from the inside
- scotland thank you danny boy trump is you the ability to lead normal lives and money the special prosecutor to
- sleepy joe biden is the best businessman in the u s senate from the best evidence is the leadership of
- apprentice tonight at 10 30am est enjoy #trump2016

numpy.random.choice

## Natural Language "Generation" ...using seed words

- crooked hillary her rallies are held in the first class funeral
- make america great again people that got rich selling out
- sleepy as he traveled with a coffee in his career tax
- sleepy joe he is the media would come in and they knew

### Google Books Ngram Viewer

Q deep learning,machine learning,artificial intelligence									
1800 - 2019 -	English (2019	e) 🗸 Cas	e-Insensitive	Smoothi	ng 🕶				
0.000400% –									
0.000350% -									
0.000300% -									
0.000250% -									
0.000200% -									
0.000150% –									
0.000100% -									
0.000050% -									
0.000000%	1820	1840	1860	1880	1900				





# **Problems with N-gram Models**

- Problem: many n-grams (and words) do not occur in the training corpus  $\rightarrow p(w) = 0$
- Use
  - Special symbol for unknown word ("unk")
  - smoothing of the estimates
  - Categories/bins of words

• Example: size of vocabulary  $|V| = 20,000 \rightarrow 8 \cdot 10^{12}$  possible trigrams

## **Smoothing of N-grams** Laplace ("add-one") Smoothing

• To prevent zero counts, add 1 to any count.

• 
$$\hat{C}(w_{i-n+1}\cdots w_i) = 1 + C(w_{i-n+1})$$

•  $p(w_i | w_{i-n+1} \cdots w_{i-1}) = \frac{\hat{C}(w_{i-n+1} \cdots w_{i-1})}{\hat{C}(w_{i-n+1} \cdots w_{i-1})}$ 

Drawback: rather high probability for non-existing (=missing) n-grams

$$\frac{\dots w_{i}}{\dots w_{i-1}} = \frac{1 + C(w_{i-n+1} \cdots w_{i})}{|V| + \sum_{w_{i} \in V} C(w_{i-n+1} \cdots w_{i})}$$

## **Smoothing of N-grams** Good-Turing Smoothing (1953)

- Idea: n-grams with the same count r in the training corpus should be assigned the same probability
- Good-Turing (GT) estimate  $r^{\star} = (r+1)$
- $\eta_r$  is the number of n-grams which occurred exactly r times
- $\hat{p}_k = r^* / N$  with  $r = N_k$
- GT requires that  $\eta_r \neq 0$  for all n-grams ( $\rightarrow$  other smoothing may be necessary)
- Frequencies not arbitrarily changed but redistributed

$$)\frac{\eta_{r+1}}{\eta_r}$$

(→ other smoothing may be necessary) **redistributed** 

## **Smoothing of N-grams Back-Off**

Idea: missing n-grams are mapped onto a lower order n-gram  $\bullet$ if C(vw) > 0if C(vw) = 0

• 
$$p_{bo}(w | v) = \begin{cases} p(w | v) \\ \beta(v) \cdot p_{bo}(w | v') \end{cases}$$

- v' is the reduced/shortened history
- Weight  $\beta(v)$  ("back-off weight") guarantees stochasticity of  $p_{bo}$

## **Smoothing of N-grams** Interpolation

- Observation: For small r, the ML estimate is very imprecise
- Mitigation: include statistics of lower order even if r > 0
- Linear interpolation  $p_{I}(w_{n} | w_{1} \cdots w_{n-1}) = \rho_{0} \frac{1}{|V|} + \rho_{1}p$

with 
$$\sum_{i} \rho_i = 1$$

Learn weights using cross-validation and expectation maximization (EM)

$$p(w_n) + \rho_2 p(w_n | w_{n-1}) + \dots$$

## **Smoothing of N-grams Other Methods**

- Kneser-Ney interpolation: non-linear interpolation of statistics
- Interpolation using max-ent  $\bullet$
- Log-linear interpolation  $\bullet$

# Using Categories

- Instead of actual types, use categories
- with  $\bigcup \mathscr{C}_k = V$ 
  - Numerals, names, cities, persons, months, etc.
  - Should be dissect; what about "Essen" (city and food)

For a categorical bi-gram:  $p(w_1 \cdots w_r)$ 

• Group words with similar functional an statistical properties  $\mathscr{C} = \{\mathscr{C}_1, \ldots, \mathscr{C}_n\}$ 

$$w_{n} = \prod_{i} p(w_{i} | \mathscr{C}(w_{i})) \cdot p(\mathscr{C}(w_{i}) | \mathscr{C}(w_{i-1}))$$



# **Cache Models**

- Context may shift: dynamically adapt statistics to the current circumstances
- Interpolate a static model  $p^{(s)}$  linearly with a cache model  $p^{(c)}$  $\bar{p}(w_i | w_{i-n+1} \cdots w_{i-1}) = \rho_c p^{(s)}(w_i | w_{i-n+1} \cdots w_{i-1}) + (1 - \rho_c) p^{(c)}(w_i | w_{i-1})$
- Cache model is continuously updated, typically low order
- Interpolation weight  $\rho_c$  dependent on cache size

### Measuring Fit How well does a model fit an observation?

 Compute the (log-)probability ("production probability") → directly impacted by sentence length

• Perplexity PP(
$$w$$
) =  $\sqrt[n]{\frac{1}{p(w_1 \cdots w_n)}}$ 

- Measures the weighted average branching factor in predicting the next word
- Lower is better

, normalized by sentence length

# Using N-gram Models for Classification

- Train an n-gram model for each class (1) and (2)
- For binary classification: compute log ratio  $s(w) = \log \frac{p^{(1)}(w)}{p^{(2)}(w)} = \begin{cases} > 0 & \text{if (1) is more likely} \\ \le 0 & \text{if (2) is more likely} \end{cases}$
- Other class could be "background class"
- Example usages:
  - Discriminate speech of healthy and people with Alzheimer's
  - Author attribution

## Limits of N-grams :-)

Recognizer output (actually impressive!)

im Bundesrat klar als Bedingung. Das heißt also Absenkung des na des des des des des eh na... ist es Alters das ist das Alter der Kinder wenn sie des Nachzugsalters. Dann kommt der fünfte Punkt und das sechste Punkt kommen sicherlich die Fragen gleich gleichgeschl gleich die gleiche schließen dann ob ich aufhöre Asylgründe schaffte außerhalb der politischen und der rassistischen Verfolgung also auch Gründe wenn aus wenn wenn andere Gründe sozusagen also aus dem Geschlecht oder ähnlichen stattfinden das er seine Frauen die irgendwie wegen ihres Frau sei es irgendwo verfolgt werden. Ob ich denen jetzt ein Asyl Asylgrund gebe...



Edmund Stoiber bei Christiansen, Jan 20, 2002

