Assignment 4,5 + Smoothing

(SNLP tutorial 4)

Vilém Zouhar, Awantee Deshpande, Julius Steuer

TODOth, TODOth May 2021

Overview

- Task, Metrics
- Differential Privacy
- Homework

Entropy

- Amount of information / compressed size in bits
- $H(p) = E[-\log(p(V))] = -\sum p(v) \log(p(v))$
- For binomial distribution highest in the middle
- For uniform distribution: $\log(W)$
- Entropy is always non-negative
- H((W,W)) = H(W)+H(W) when statistically independent p(w1,w2) = p(w1)p(w2)
- Conditional entropy: $H(X|Y) = -\sum p(x,y) \log p(x|y)$

Kullback-Leibler Divergence

- $D(p||q) = \sum p_i \log p_i / q_i$
- Not symmetric
- Non-negative
- How many extra bits if we use bad encoding
- Cross-entropy: $-\sum p_i \log q_i$

Code

- Mapping of word to a finite string of a D-nary alphabet
- Prefix code
- $\sum D^{-l_i} \leq 1$
-
  - Krafts inequality
-
  - true for prefix codes
-
  - for every length distribution satisfying this, there exists a prefix code
- Expected length: $\sum l_i p(w_i)$
- Optimal length: $-\log_D p(w_i)$

Correlation Function

- $p_d(w1, w2)/(p(w1)p(w2))$

OOV words

Corpus

- Train set:
  🍏 🍏 🍏 🍆 🍏 🍌 🍌 🍒 🍏 🍆 🍌 🍌 🍒 🍌 🍏 🍆
- Test set:
  🍫 🍏 🍟 🍌 🍏 🍆 🍆 🍌 🍒 🍟 🍏 🍏

Accumulate counts

- 🍏 6 🍌 5 🍆 3 🍒 2
- 🍏 4 🍌 2 🍟 2 🍆 2 🍒 1 🍫 1

OOV words

- What about 🍫 and 🍟?
- OOV rate: $2 + 1/4 + 2 + 2 + 1 + 1 + 1 = 27\%$

Additive smoothing (add-α-smoothing)

Unigrams

- Add zero counts to frequency table

🍏 6 🍌 5 🍆 3 🍒 2 🍟 0 🍫 0

- Increase all counts by $\alpha = 1$

🍏 6+1 🍌 5+1 🍆 3+1 🍒 2+1 🍟 0+1 🍫 0+1

- Divide by $N = 22$

🍏 0.32 🍌 0.27 🍆 0.18 🍒 0.13 🍟 0.05 🍫 0.05

Perplexity

- Relative frequencies on test corpus:

🍏 0.33 🍌 0.17 🍟 0.17 🍆 0.17 🍒 0.08 🍫 0.08

- $\text{PP} = 2^{(0.33 \cdot 0.32 + 0.27 \cdot 0.17 + 0.18 \cdot 0.17 + 0.13 \cdot 0.17 + 2 \cdot (0.05 \cdot 0.08))} = 1.4$

Additive smoothing: Bigrams

Recall the additive smoothing formula for unigrams:

$$p_{smoothed}(w_i) = \frac{C(w_i) + \alpha}{N + \alpha|V|} \tag{1}$$

- What is N? What is V?

Remember from Assignment 2 that:

$$p(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})} \tag{2}$$

- Smoothe the bigram count: $C(w_{i-1}, w_i) \rightarrow C(w_{i-1}, w_i) + \alpha$
- Normalization: $p_{smoothed}(w_i|w_{i-1}) = \frac{C(w_{i-1},w_i)+\alpha}{?}$

Additive smoothing: Bigrams

Corpus

🍏 🍏 🍏 🍆 🍏 🍌 🍌 🍒 🍏 🍆 🍌 🍌 🍒 🍌 🍏 🍆

Bigrams: 🍏🍏, 🍏🍏, 🍏🍆, 🍆🍏, . . . , 🍏🍆, 🍆🍏 $\leftarrow$ circular bigram!

Bigrams: AA, AA, AE, EA, . . . , AE, EA

Additive smoothing: Bigrams: bigram counts

- Collect bigram counts & condtional probabilities for history A

Bigram	$C(w_i, w_{i-1})$	$C(w_{i-1})$	$\frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$
AE	3	6	$1/2$
AA	2	6	$1/3$
AB	1	6	$1/6$

Additive smoothing: Bigrams: add alpha

- We encounter an unknown bigram *AF*

Bigram	$C_\alpha(w_{i-1}, w_i)$	$C_\alpha(w_{i-1})$	$\frac{C_\alpha(w_{i-1},w_i)}{C_\alpha(w_{i-1})}$
AE	3+1	6+1	4/7
AA	2+1	6+1	3/7
AB	1+1	6+1	2/7
$\rightarrow$ AF	0+1	6+1	1/7

- Not a probabilitiy distribution!
- Solution: We need to adjust the divisor a tiny bit. But how tiny?

Additive smoothing: Bigrams: normalization

- add $\alpha 3$ to history count!
- Pretend that we have seen the history $|V| = 3$ times more.

Bigram	$C_\alpha(w_{i-1}) + \alpha\|V\|$	$\frac{C_\alpha(w_{i-1},w_i)}{C_\alpha(w_{i-1})+\alpha\|V\|}$
AE	$7+3$	$4/10$
AA	$7+3$	$3/10$
AB	$7+3$	$2/10$
$\rightarrow$ AF	$7+3$	$1/10$

- Now the probabilities sum up to 1: $4/10 + 3/10 + 2/10 + 1/10 = 1$

Additive smoothing: Bigrams: normalization

- We encounter another n-gram AD
- What is $|V|$ now?

Bigram	$C_\alpha(w_{i-1}) + \alpha\|V\|$	$\frac{C_\alpha(w_{i-1},w_i)}{C_\alpha(w_{i-1})+\alpha\|V\|}$
AE	$7+4$	$4/11$
AA	$7+4$	$3/11$
AB	$7+4$	$2/11$
$\rightarrow$ AF	$7+4$	$1/11$
$\rightarrow$ AD	$7+4$	$1/11$

- $C_\alpha(A)$ is constant
- Probabilities sum up to 1: $4/11 + 3/11 + 2/11 + 1/11 + 1/11 = 1$

Additive smoothing: Bigrams: general case

- General formula for smoothed bigram Probabilities:

$$p(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + \alpha}{C(w_{i-1}) + \alpha|V|} \tag{3}$$

- What is V?
- $|V|$ = Number of bigram **types** starting with w_{i-1}

$$p(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + \alpha}{C(w_{i-1}) + \alpha|V_{(w_{i-1},\bullet)}|} \tag{4}$$

- For n-grams of length n:

$$p(w_i|w_{i-1} : w_{i-n+1}) = \frac{C(w_{i-n+1} : w_i) + \alpha}{C(w_{i-n+1} : w_{i-1}) + \alpha|V_{(w_{i-n+1}:w_{i-1},\bullet)}|} \tag{5}$$

Kneser-Ney Smoothing

TODO

- absolute discounting

Cross-Validation

TODO

Estimating LOO Parameters

TODO ??

Laplace Smoothing

- add epsilon

TODO

Linear Discounting

- linear interpolation

Good-Turing Discounting

TODO

Count Trees

- remove infrequent nodes

TODO

Privacy

TODO differential privacy

Resources

1. UdS SNLP Class, WSD: https://teaching.lsv.uni-saarland.de/snlp/
2. Classical Statistical WSD: https://www.aclweb.org/anthology/P91-1034.pdf
3. n-gram count trees: http://ssli.ee.washington.edu/WS07/notes/ngrams.pdf