CS565: Intelligent Systems and Interfaces

Lecture: Language Modeling
Estimating Parameters of N-gram models

3rd Feb, 2016
Semester: Jan – May 2016

Ashish Anand
IIT Guwahati

Recap and Moving Forward

- In the last lecture
  - Language Modeling
    - Definition
    - N-gram Models
    - Parameter Estimation
- Moving Forward
  - Better estimators: Smoothing Techniques

MLE of N-gram models

- Unigram

$$p_{ml}(w_i) = \frac{c(w_i)}{\sum c(w_i)}$$

- Bigram

$$p_{ml}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- Trigram

$$p_{ml}(w_i|w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

Problem with MLE

- Works well if test corpus is very similar to training, which is not generally the case.
- Training Set

  …… denied the allegations

  …… denied the reports

  …… denied the claims

  …… denied the request
- Test Set

  …. denied the offer

  …. denied the loan

P("offer" | denied the) = 0

Smoothing Techniques

Simplest Approach: Additive Smoothing

- Add-1 Smoothing

$$p_{ml}(w_i|w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i) + 1}{c(w_{i-2}, w_{i-1}) + |\mathcal{V}|}$$

- Generalized version

$$p_{ml}(w_i|w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i) + \delta}{c(w_{i-2}, w_{i-1}) + \delta|\mathcal{V}|}$$

What’s wrong with additive smoothing

- Gale and Church, 1990, *Estimation procedure for language context: poor estimates are worse than none.* In *COMPSTAT*, Proceedings in Computational statistics

- Gale and Church, 1994, *What’s wrong with adding one?* Corpus-Based Research into Language.

Take the help of lower order models

- Bigram Example
  - $c(w_1, w_2) = 0 = c(w_1, w_2')$
  - $p_{add} (w_2 \mid w_1) = p_{add} (w_2' \mid w_1)$
  - Lets assume $p(w_2') < p(w_2)$
  - We should expect $p_{add} (w_2 \mid w_1) > p_{add} (w_2' \mid w_1)$

Take the help of lower order models

- Linear Interpolation Models
- Discounting Models

Linear Interpolation Model

- Bigram model $p(w_i|w_{i-1})$

$$p_{int}(w_i|w_{i-1}) = \lambda p_{ml}(w_i|w_{i-1}) + (1-\lambda)p_{ml}(w_i),$$

Where $0 \leq \lambda \leq 1$

- Trigram model

$$p_{int}(w_i|w_{i-2},w_{i-1})$$
$$= \lambda_1 \times p_{ml}(w_i|w_{i-2},w_{i-1}) + \lambda_2 \times p_{ml}(w_i|w_{i-1}) + \lambda_3 \times p_{ml}(w_i),$$

Linear Interpolation Model

Verify $p_{int}(w_i|w_{i-2}, w_{i-1})$ is probability distribution.

i.e., $\sum p_{int}(w_i|w_{i-2}, w_{i-1}) = 1$

Estimating λ values

- Use of validation or development or held-out data

- $c'(w_1,w_2,w_3)$:= Number of occurrences of $w_1w_2w_3$ in the ***validation data***

- Maximum likelihood estimation

$$L(\lambda_1, \lambda_2, \lambda_3) = \sum_{w_1,w_2,w_3} c'(w_1, w_2, w_3) \log p_{int}(w_3|w_1, w_2)$$

s.t. constraints on λ values.

Allowing λ to vary

- Objective: vary the weight as per the count that is being conditioned.
- For trigram, we are conditioning on bigrams.
- Approach – “Bucketing”
- Example

Discounting Method

- Collins Slide

More on Smoothing Techniques

- An Empirical Study of Smoothing Techniques for Language Modeling, *S Chen, and J Goodman*, 1998.

- Generalized versions
  - Interpolation Techniques
  - Discounting Methods

Evaluating Language Models: Perplexity

- Given a test data of *m* sentences: $s_1, s_2, \ldots\ldots, s_m$
- Probability of a sentence under this model p(s_i)
- Log-Probability of all sentences: $\log \prod p(s_i) = \sum \log p(s_i)$
- Perplexity = 2^{-l} , where l = 1/M($\sum \log p(s_i)$)

- Smaller the value of perplexity, better the language model is.