Language Modeling
"How to construct LM?"

Koichiro Yoshino

Kyoto University
Media Archiving Research Laboratory

Generative Approach

- Automatic speech recognition model is based on generative approach (Noisy channel model).
- Noisy channel model is used for OCR, IME etc···

- $\boldsymbol{W} = \mathbf{argmax} \boldsymbol{P}(\boldsymbol{W}|\boldsymbol{S}) = \mathbf{argmax} \underbrace{\boldsymbol{P}(\boldsymbol{W})}_{\text{LM}} \underbrace{\boldsymbol{P}(\boldsymbol{S}|\boldsymbol{W})}_{\text{AM}}$

Language Model $P(W)$

- $w_1, w_2, \dots, w_n \in W$
  - (word sequence W consisting of with n words)
- $P(W) = P(w_1)P(w_2|w_1)\dots P(w_n|w_1, \dots, w_{n-1})$

- $P(\text{You can't eat grilled eel without sprinking sansho.})$
  $= P(\text{you})P(\text{can't}|\text{you})\dots P(.|\text{you}, \dots, \text{sansho})$
  - This is word unit case.

- N-gram is approximate approach for the prob. definition.
  - Uni-gram: $P(W) = \prod_i P(w_i)$
  - Bi-gram: $P(W) = \prod_i P(w_i|w_{i-1})$
    - (every word's probability depends on the previous word)

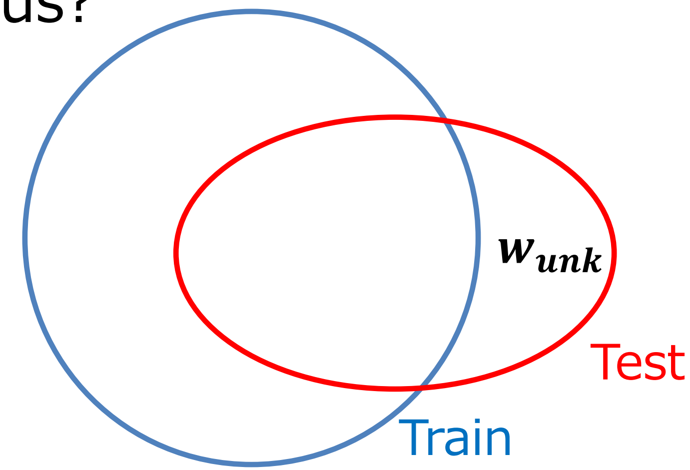
N-gram

- N-gram models give the conditional probability of a word.
  - N-1 words are given as a condition.

- Uni-gram (no condition).
  - $\boldsymbol{P_{uni}(w_i) = \frac{C(w_i)}{\sum_i C(w_i)}}$
- Bi-gram (conditioned on the previous word).
  - $\boldsymbol{P_{bi}(w_i) = P(w_i|w_{i-1}) = \frac{C(w_i, w_{i-1})}{C(w_{i-1})}}$
- Tri-gram (conditioned on the previous 2 words).
  - $\boldsymbol{P_{tri}(w_i) = P(w_i|w_{i-1}, w_{i-2}) = \frac{C(w_i, w_{i-1}, w_{i-2})}{C(w_{i-1}, w_{i-2})}}$

Unknown word problem

- Word based LM has unknown word problem.
  - If you can't find word $\boldsymbol{w_{unk}}$ in the training corpus, how to define $\boldsymbol{P(w_{unk})}$ in the test corpus?



1. Give unknown words probability with some smoothing method.
   – Ex) add-1 smoothing.

2. Use unknown word model based on character LM.
   – Use char. LM when the observed word is unknown.

Character LM

- What would you do to use character unit?
  - $\boldsymbol{P}(\text{You can't eat grilled eel without sprinking sansho.})$
    $= \boldsymbol{P}(\text{y})\boldsymbol{P}(\text{o}|\text{y})\boldsymbol{P}(\text{u}|\text{yo}), \dots, \boldsymbol{P}(.|\text{y}, \text{o}, \text{u}, \text{c}, \text{a}, \text{n}, ', \text{t}, \dots, .)$

- Char. based uni-gram LM is defined as $\boldsymbol{P}(\boldsymbol{C}) = \prod_{\boldsymbol{i}} \boldsymbol{P}(\boldsymbol{c_i})$

- However, what should we do when we find an unobserved character $\boldsymbol{c_u}$?
  - Use zero-gram model.

Char. Zero-gram Model

- Character sets are defined in a character encoding scheme (94 characters for English).
- In a char. zero-gram, every character has equivalent probability (=1/94).
  - Japanese has 6878 characters in JIS X 0208.
- $\boldsymbol{P}(\boldsymbol{W}) = \prod_{\boldsymbol{i}} \left(\frac{\mathbf{1}}{\mathbf{94}}\right)^{\boldsymbol{i}}$

Char. based unknown word model

- You use char. based probability when you find an unknown word $\boldsymbol{w_{unk}}$.
- Char. based probability is calculated with char. N-gram model.
- If you find an unknown character $\boldsymbol{c_{unk}}$ (unobserved character in training set), you use char. zero-gram.

Word N-gram

$$P(W) = \prod_i P(w_i | w_{i-1}, \dots, w_{i-(n-1)})$$

$$\boxed{P(w_{unk}) = \prod_k P(c_k)}$$

Char. N-gram

$$P(c_k) = \prod_k P(c_k | c_{k-1}, \dots, c_{k-(n-1)})$$

$$\boxed{P(c_{unk}) = P_{zero}(c_{unk})}$$

Char. zero-gram

$$P_{zero}(c_{unk}) = \frac{1}{94}$$

Coverage

- Coverage is the percentage of words which are covered by the training-set.
  - Count of all words. Not kinds of word.
  - Lower coverage means there is a high risk of unknown words.
    - Ex). Tri-gram is more space than uni-gram.

- N-gram coverage varies with "N".
  - Larger N means lower coverage.
    - $\boldsymbol{P}(\mathrm{eat}|\mathrm{you\ can't})$ has better prediction accuracy and lower coverage than $\boldsymbol{P}(\mathrm{eat}|\mathrm{can't})$.

Interpolation

- High order N-gram models improve prediction accuracy but increase the risk of uncovered words.

- In an interpolation model, a weighted probability is calculated from both higher and lower order N-grams.
  - Ex). Interpolation of bi-gram and uni-gram
    - $\boldsymbol{P(w_i) = \alpha P(w_i|w_{i-1}) + (1-\alpha)\ P(w_i)}$
  - Ex). Interpolation of char. bi-gram and zero-gram
    - $\boldsymbol{P(c_i) = \beta P(c_i|c_{i-1}) + (1-\beta) P_{zero}(c_{unk})}$

- Other smoothing method
  - back-off, witten-bell, etc.

Perplexity

- How can we measure a language model's prediction accuracy?

- Entropy (H) is used for model adequacy.
  - $\boldsymbol{H = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(w_i)}$
  - Entropy is necessary bits for the model.
  - LM can compress texts down to the value of entropy.
    - Default is zero-gram (=1/94).

- Perplexity is the average number of candidate words to choose from.
  - $\boldsymbol{PP = 2^H}$

Implementation

1. Count the different types of characters in the training-set.

2. Calculate the character coverage.

3. Construct a character uni-gram model and calculate the test-set perplexity (use interpolation with the zero-gram model. $\boldsymbol{\beta = 0.1}$).

4. Construct a word uni-gram model and calculate the test-set perplexity (skip any unknown words).

Implement

5. Drop low-frequency word probabilities and use their probabilities for unknown word probabilities (cutoff=1). (Use the uni-gram model from step 4. and the character-based model from step 3. as the unknown word model)
   – $\boldsymbol{P(w_{unk}) = \sum_{w(C(w)=1)} P(w)}$

6. Construct a word bi-gram model interpolate with the unigram and the character-based unknown word model. ($\boldsymbol{\alpha = 0.5}$)

7. Change the $\boldsymbol{\alpha}$ by increments of 0.1 and find the best value for $\boldsymbol{\alpha}$.

Tips

- Use log for probability calculations.
  - ignore rounding errors
  - $\mathbf{log}\boldsymbol{P}(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{y}) = \boldsymbol{logP}(\boldsymbol{x}) + \boldsymbol{logP}(\boldsymbol{y})$
  - $\mathbf{log}\,\frac{\boldsymbol{P}(\boldsymbol{x})}{\boldsymbol{P}(\boldsymbol{y})} = \boldsymbol{logP}(\boldsymbol{x}) - \boldsymbol{logP}(\boldsymbol{y})$

- Use a start of sentence symbol for bi-gram modeling.
  - If the word $\boldsymbol{w_1}$ is the first word in the sentence, bi-gram based probability is calculated as $\mathrm{P}(\boldsymbol{w_1}|\boldsymbol{w_{start}})$.