Oberseminar

25.07.2013

Introduction to Kneser-Ney Smoothing on Top of Generalized Language Models for Next Word Prediction

Martin Körner

Content

- Introduction
- Language Models
- Generalized Language Models
- Smoothing
- Progress
- Summary

Content

Introduction: Motivation

- Next word prediction: What is the next word a user will type?

- Use cases for next word prediction:

  - Augmentative and Alternative Communication (AAC)

  - Small keyboards (Smartphones)

Introduction to next word prediction

- How do we predict words?

1. Rationalist approach
   - Manually encoding information about language
   - “Toy” problems only

2. Empiricist approach
   - Statistical, pattern recognition, and machine learning methods applied on corpora
   - Result: Language models

Content

- Introduction
- **Language Models**
- Generalized Language Models
- Smoothing
- Progress
- Summary

Language models in general

- Language model: How likely is a sentence s?
  → Probability distribution: $P(s)$
- Calculate $P(s)$ by multiplying conditional probabilities
- Example:

$$P(\text{If you're going to San Francisco , be sure } \dots)$$
$$=$$
$$P(\text{you're} \mid \text{If}) * P(\text{going} \mid \text{If you're}) *$$
$$P(\text{to} \mid \text{If you're going}) * P(\text{San} \mid \text{If you're going to}) *$$
$$P(\text{Francisco} \mid \text{If you're going to San}) * \cdots$$

→ Empirical approach would fail

Conditional probabilities simplified

- Markov assumption [JM80]:
  - Only the last *n-1* words are relevant for a prediction
  - Example with *n=5:*

$$P(\text{sure} \mid \text{If you're going to San Francisco , be})$$
$$\approx P(\text{sure} \mid \text{San Francisco , be})$$

Counts as a word

Definitions and Markov assumption

- *n*-gram: Sequence of length *n* with a count
  - E.g.: 5-gram:

$$\text{If you're going to San} \qquad 4$$

- Sequence naming:

$$w_1^{i-1} := w_1\ w_2\ \ldots w_{i-1}$$

- Markov assumption formalized:

$$P\left(w_i \middle| w_1^{i-1}\right) \approx P\left(w_i \middle| \underbrace{w_{i-n+1}^{i-1}}_{n\text{-1 words}}\right)$$

Formalizing next word prediction

- Instead of $P(s)$:
  - Only one conditional probability $P\left(w_i \middle| w_{i-n+1}^{i-1}\right)$
    - Simplify $P\left(w_i \middle| w_{i-n+1}^{i-1}\right)$ to $P\left(w_n \middle| w_1^{n-1}\right)$

n-1 words *n*-1 words

Conditional probability with Markov assumption

$$\mathrm{NWP}(w_1^{n-1}) = \arg\max_{w_n \in W} \left(P\left(w_n \middle| w_1^{n-1}\right)\right)$$

Set of all words in the corpus

→ How to calculate the probability $P\left(w_n \middle| w_1^{n-1}\right)$?

How to calculate $P(w_n|w_1^{n-1})$

- The easiest way:
  - Maximum likelihood:

$$P_{\mathrm{ML}}(w_n|w_1^{n-1}) = \frac{c(w_1^n)}{c(w_1^{n-1})}$$

  - Example:

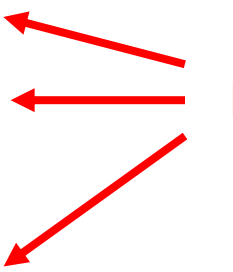
$$P(\text{San} \mid \text{If you're going to}) = \frac{c(\text{If you're going to San})}{c(\text{If you're going to})}$$

Content

- Introduction
- Language Models
- **Generalized Language Models**
- Smoothing
- Progress
- Summary

Intro Generalized Language Models (GLMs)

- Main idea:
  - Insert wildcard words ($*$) into sequences
- Example:
  - Instead of $P(\text{San} \mid \text{If you're going to})$:
    - $P(\text{San} \mid \text{If} \quad * \quad * \quad *)$
    - $P(\text{San} \mid \text{If} \quad * \quad * \quad \text{to})$
    - $P(\text{San} \mid \text{If} \quad * \quad \text{going} \quad *)$
    - $P(\text{San} \mid \text{If} \quad * \quad \text{going} \quad \text{to})$
    - $P(\text{San} \mid \text{If you're} \quad * \quad *)$
    - …



Length: 5, Wildcard words: 2

→ Aggregate results

  - Separate different types of GLMs based on:
    1. Sequence length
    2. Number of wildcard words

Why Generalized Language Models?

- Data sparsity of *n*-grams
  - "If you're going to San" is seen less often than for example "If * * to San"

→ Question: Does that really improve the prediction?

  - Result of evaluation: Yes

… but we should use smoothing for language models

Content

- Introduction
- Language Models
- Generalized Language Models
- **Smoothing**
- Progress
- Summary

Smoothing

- Problem: Unseen sequences

  → Try to estimate probabilities of unseen sequences

  - Probabilities of seen sequences need to be reduced

- Two approaches:

  1. Backoff smoothing
  2. Interpolation smoothing

Backoff smoothing

- If sequence unseen: use shorter sequence
  - E.g.: if $P(\text{San} \mid \text{going to}) = 0$ use $P(\text{San} \mid \text{to})$

Higher order probability

$$P_{back}\left(w_n \middle| w_i^{n-1}\right) = \begin{cases} \tau\left(w_n \middle| w_i^{n-1}\right) & if\ c(w_i^n) > 0 \\ \gamma * P_{back}\left(w_n \middle| w_{i+1}^{n-1}\right) & if\ c(w_i^n) = 0 \end{cases}$$

Weight

Lower order probability (recursive)

Interpolated Smoothing

- Always use shorter sequence for calculation

$$P_{inter}\left(w_n \middle| w_i^{n-1}\right) = \underbrace{\tau\left(w_n \middle| w_i^{n-1}\right)}_{\text{Higher order probability}} + \underbrace{\gamma}_{\text{Weight}} * \underbrace{P_{inter}\left(w_n \middle| w_{i+1}^{n-1}\right)}_{\text{Lower order probability (recursive)}}$$

- Seems to work better than backoff smoothing

Kneser-Ney smoothing [KN95] intro

- Interpolated smoothing
- Idea: Improve lower order calculation
- Example: Word visiting unseen in corpus

$P(\text{Francisco} \mid \text{visiting}) = 0$

→ Normal interpolation: $0 + \gamma * P(\text{Francisco})$

$P(\text{San} \mid \text{visiting}) = 0$

→ Normal interpolation: $0 + \gamma * P(\text{San})$

Result: Francisco is as likely as San at that position

Is that correct?

→ Difference between Francisco and San?

Answer: Number of different contexts

Kneser-Ney smoothing idea

- For lower order calculation:
  - Don't use $c(w_n)$
  - Instead: Number of different bigrams the word completes:

$$N_{1+}(\bullet\, w_n) := |\{w_{n-1} : c(w_{n-1}^n) > 0\}|$$

Count

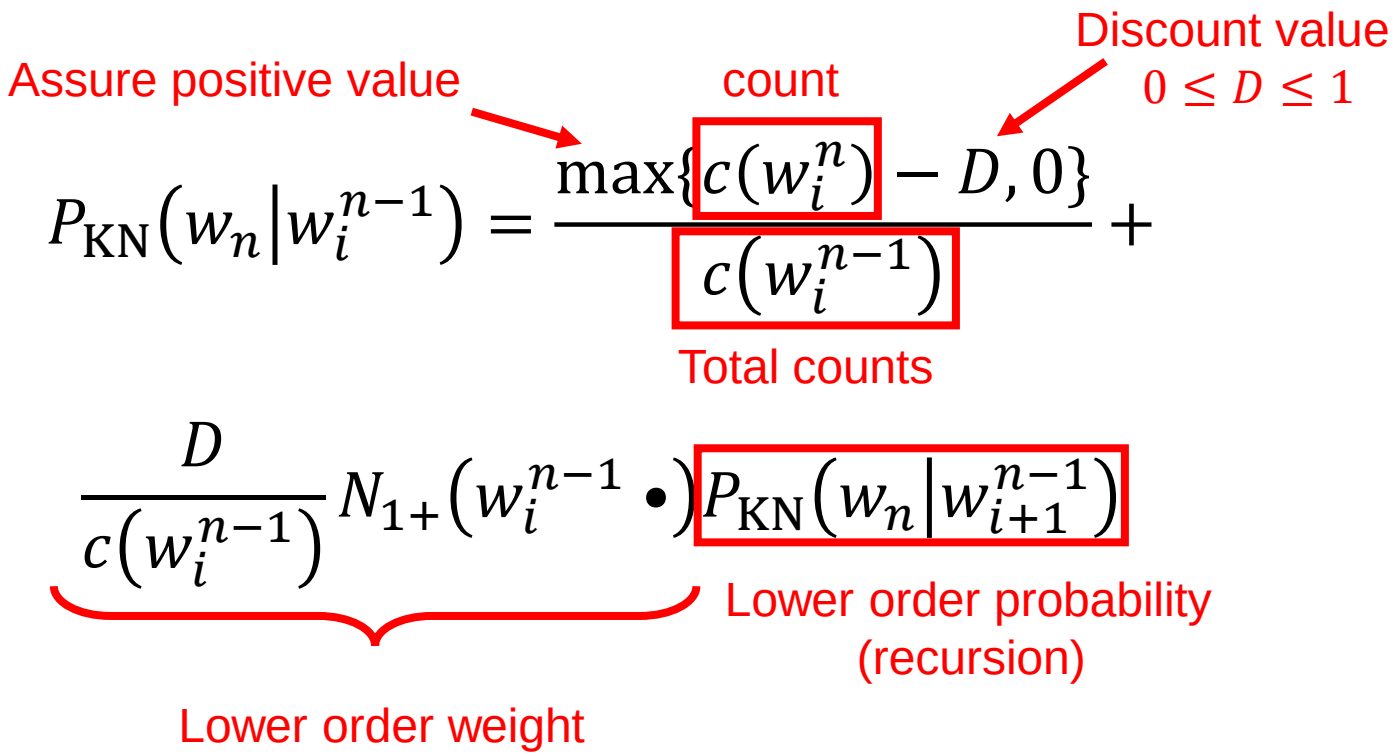
  - Or in general:

$$N_{1+}(\bullet\, w_{i+1}^n) = |\{w_i : c(w_i^n) > 0\}|$$

- In addition:
  - $N_{1+}\left(\bullet\, w_{i+1}^{n-1} \bullet\right) = \sum_{w_n} N_{1+}(\bullet\, w_{i+1}^n)$
  - $N_{1+}\left(w_i^{n-1}\, \bullet\right) = |\{w_n : c(w_i^n) > 0\}|$

Kneser-Ney smoothing equation (highest)

- Highest order calculation:



Kneser-Ney smoothing equation

- Lower order calculation:

Assure positive value

Continuation count

Discount value

$$P_{\mathrm{KN}}\left(w_n \middle| w_i^{n-1}\right) = \frac{\max\{N_{1+}(\bullet\, w_i^n) - D, 0\}}{N_{1+}\left(\bullet\, w_i^{n-1}\, \bullet\right)} + \frac{D}{N_{1+}\left(\bullet\, w_i^{n-1}\, \bullet\right)} N_{1+}\left(w_i^{n-1}\, \bullet\right) P_{\mathrm{KN}}\left(w_n \middle| w_{i+1}^{n-1}\right)$$

Total continuation counts

Lower order probability (recursion)

Lower order weight

- Lowest order calculation: $P_{\mathrm{KN}}(w_n) = \frac{N_{1+}(\bullet w_i^n)}{N_{1+}(\bullet w_i^{n-1} \bullet)}$

Modified Kneser-Ney smoothing [CG98]

- Different discount values for different absolute counts
- Lower order calculation:

$$P_{\mathrm{KN}}\left(w_n \middle| w_i^{n-1}\right) = \frac{\max\{N_{1+}(\bullet\, w_i^n) - D(c(w_i^n)), 0\}}{N_{1+}\left(\bullet\, w_i^{n-1}\, \bullet\right)} + \frac{D_1 N_1\left(w_i^{n-1}\, \bullet\right) + D_2 N_2\left(w_i^{n-1}\, \bullet\right) + D_{3+} N_{3+}\left(w_i^{n-1}\, \bullet\right)}{N_{1+}\left(\bullet\, w_i^{n-1}\, \bullet\right)} P_{\mathrm{KN}}\left(w_n \middle| w_{i+1}^{n-1}\right)$$

- State of the art (since 15 years!)

Smoothing of GLMs

- We can use all smoothing techniques on GLMs as well!

- Small modification:

  E.g: $P(\text{San} \mid \text{If} \quad * \quad \text{going} \quad *)$

  Lower order sequence :

  – Normally: $P(\text{San} \mid * \quad \text{going} \quad *)$

  – Instead use $P(\text{San} \mid \text{going} \quad *)$

Content

- Introduction
- Language Models
- Generalized Language Models
- Smoothing
- **Progress**
- Summary

Progress

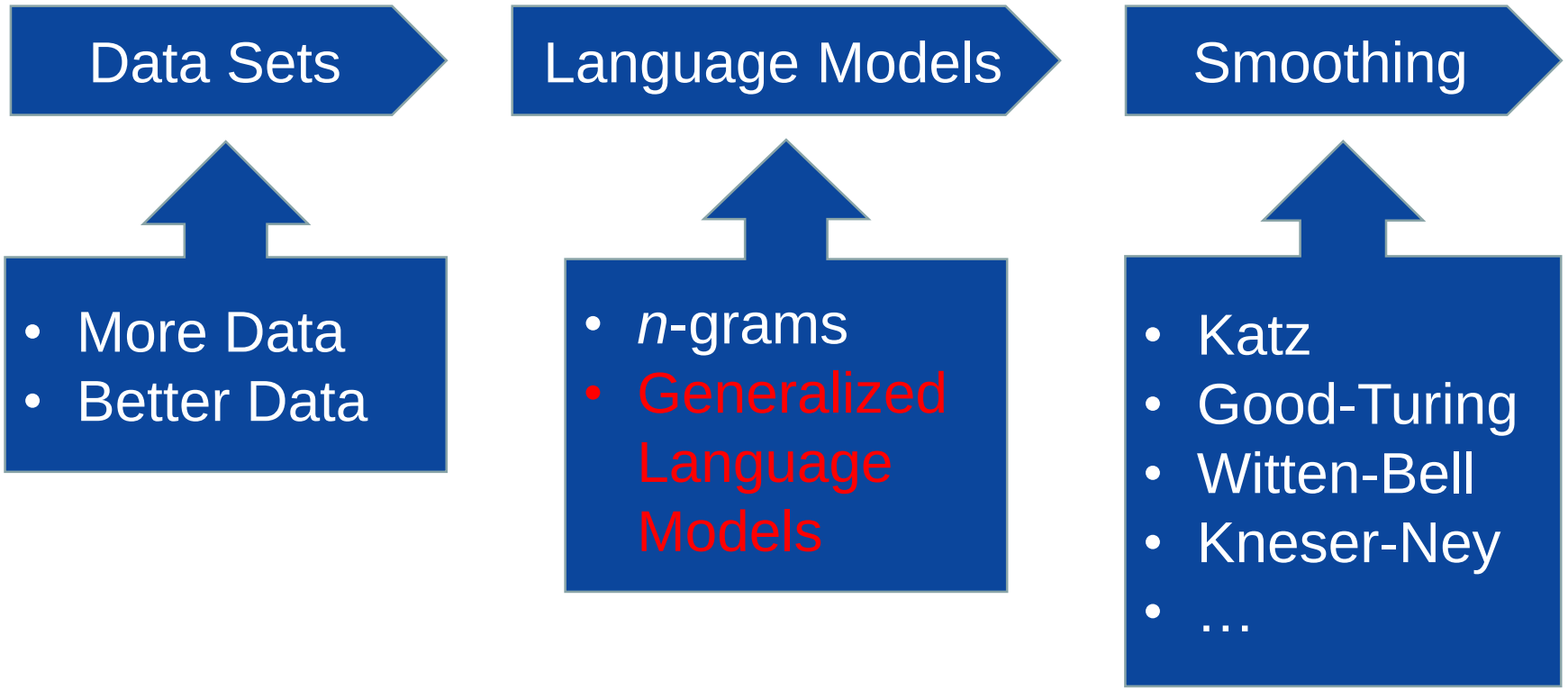
- Done Yet:
  - Extract text from XML files
  - Building GLMs
  - Kneser-Ney and modified Kneser-Ney smoothing
  - Indexing with MySQL

- ToDo’s
  - Finish evaluation program
  - Run evaluation
  - Analyze results

Content

- Introduction
- Language Models
- Generalized Language Models
- Smoothing
- Progress
- **Summary**

Summary


Data Sets
Language Models
Smoothing
More Data
Better Data
n-grams
Generalized Language Models
Katz
Good-Turing
Witten-Bell
Kneser-Ney
…

Thank you for your attention!

Questions?

- **Images:**
  - Wheelchair Joystick (Slide 4): http://i01.i.aliimg.com/img/pb/741/422/527/527422741_355.jpg
  - Smartphone Keyboard (Slide 4): https://activecaptain.com/articles/mobilePhones/iPhone/iPhone_Keyboard.jpg
- **References:**

  - **[CG98]:** Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical report, Technical Report TR-10-98, Harvard University, August, 1998.
  - **[JM80]:** F. Jelinek and R.L. Mercer. Interpolated estimation of markov source parameters from sparse data. In Proceedings of the Workshop on Pattern Recognition in Practice, pages 381–397, 1980.
  - **[KN95]:** Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, volume 1, pages 181–184. IEEE, 1995.