# Bayesian Class-based Language Models

Yi Su
Nuance Communications, Inc.

## Introduction

- What are class-based language models (CLMs)?
  - A large family of language models which use word classes
  - "Similar words appear in similar context"
  - Most of them maintain a *hard-clustering* assumption
    - One word belongs to only one class
    - For theoretical and computational simplicity
- We propose a Bayesian formulation of the CLM
  - Naturally supports *soft-clustering*
    - One word belongs to many classes with probabilities
  - "Latent Hierarchical Pitman-Yor Allocation for Language Modeling"
    - History as "document" in Latent Dirichlet Allocation (LDA) parlance
    - Cares more about language modeling (perplexity) than topic modeling (word classes)

## Class-based Language Models (CLMs)

- Definition

$$P(w \mid h) = P(c(w) \mid h)P(w \mid c(w), h) \tag{1}$$

- Finding classes
  - Agglomerative clustering (Brown et al, 1992)
  - Exchange-based clustering (Martin et al, 1998)
- Random clustering (Emami and Jelinek, 2005)
  - Averaging many CLMs with word classes derived from randomly initialized exchange-based clustering gives better results.

$$P(w \mid h) = \frac{1}{K}\sum_{k=1}^{K} P_k(c_k(w) \mid h)P_k(w \mid c_k(w), h), \tag{2}$$

## Soft-clustering Class-based Language Models (SCLMs)

- Definition

$$P(w \mid h) = \sum_{c \in \mathcal{C}} P(c \mid h)P(w \mid c, h) \tag{3}$$

- Random sampling
  - Let $A$ denote a class assignment of all words in the training text.
  - Given $A$, class model and word model are just regular LMs.

$$\begin{aligned} P(w \mid h) &= \sum_{A \in \mathcal{A}} P(A) \sum_{c \in \mathcal{C}} P_A(c \mid h)P_A(w \mid c, h) \\ &\approx \frac{1}{K}\sum_{k=1}^{K}\sum_{c \in \mathcal{C}} P_{A_k}(c \mid h)P_{A_k}(w \mid c, h) \end{aligned} \tag{4}$$

## Soft-clustering Class-based Language Models: Inference

- Model does not subscribe itself to any inference algorithm.
- Collapsed Gibbs sampler comes natural to this model:

$$P(c_i = j \mid \mathbf{c}_{\neg i}, \mathbf{w}) \propto P(c_i = j \mid \mathbf{c}_{\neg i}, \mathbf{w}_{\neg i}) \cdot P(w_i \mid c_i = j, \mathbf{c}_{\neg i}, \mathbf{w}_{\neg i}) \tag{5}$$

- Two terms on the right hand side can be computed with "leave-one-out" versions of class and word models, respectively.
- Taking multiple samples is trivially parallelizable.

## Fully Bayesian Formulation

- Smoothing is to frequentists as prior is to Bayesians.
  - Hierarchical Pitman-Yor (HPY) prior is the Bayesian counterpart of Kneser-Ney (KN) smoothing. (Teh, 2006)
  - Or Kneser-Ney smoothing is a frequentist approximation of HPY.
- Soft-clustering Class-based Hierarchical Pitman-Yor LMs (SCHPYLMs)
  - Uses HPY LM in place of KN LM in the SCLM
  - A fully Bayesian formulation of the CLM
  - Plug in a Gibbs sampler of the HPY LM to get a sampler of the whole model

## Fully Bayesian Formulation: Generative Process

1. Choose parameters $d_j$ and $\theta_j$ for $j \in \{0, 1, \cdots, n-1\}$:

$$\begin{aligned} d_j &\sim \text{Beta}(1, 1) \\ \theta_j &\sim \text{Gamma}(1, 1) \end{aligned} \tag{6}$$

2. Choose parameters $e_j$ and $\mu_j$ for $j \in \{0, 1, \cdots, n\}$:

$$\begin{aligned} e_j &\sim \text{Beta}(1, 1) \\ \mu_j &\sim \text{Gamma}(1, 1) \end{aligned} \tag{7}$$

3. For every word $w_i$ and its history $h_i = w_{i-1} \cdots w_{i-n+1}$:

3.1 Generate $c_i$:

$$\begin{aligned} G_\phi(c) &\sim \text{PY}(d_0, \theta_0, G_0(c)) \\ G_{w_{i-1}}(c) &\sim \text{PY}(d_1, \theta_1, G_\phi(c)) \\ G_{w_{i-1}w_{i-2}}(c) &\sim \text{PY}(d_2, \theta_2, G_{w_{i-1}}(c)) \\ &\cdots \\ c_i \mid h_i &\sim \text{Mult}(G_{h_i}(c)), \end{aligned} \tag{8}$$

3.2 Generate $w_i$:

$$\begin{aligned} H_\phi(w) &\sim \text{PY}(e_0, \mu_0, H_0(w)) \\ H_{c_i}(w) &\sim \text{PY}(e_1, \mu_1, H_\phi(w)) \\ H_{c_i w_{i-1}}(w) &\sim \text{PY}(e_2, \mu_2, H_{c_i}(w)) \\ H_{c_i w_{i-1} w_{i-2}}(w) &\sim \text{PY}(e_3, \mu_3, H_{c_i w_{i-1}}(w)) \\ &\cdots \\ w_i \mid c_i, h_i &\sim \text{Mult}(H_{c_i h_i}(w)). \end{aligned} \tag{9}$$

## Experiments: Perplexity

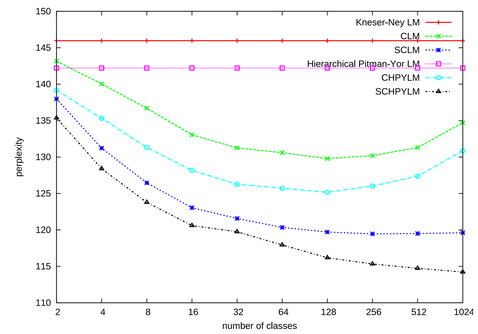


Figure: Perplexity as function of the number of classes

- Setup
  - 1M words Wall Street Journal (WSJ)
  - 10K vocabulary, 3-grams
  - 100 samples per experiment
  - 800 iterations of burn-in
- Results
  - Perplexity decreases then increases, as expected
  - Soft-clustering beats hard-clustering
  - HPY beats KN

## Experiments: Word Error Rate (WER)

- Setup
  - Lattice-rescoring
  - IBM 2004 Rich Transcription Conversational Telephony Speech (RT-04 CTS) system
  - 20M words Fisher data
  - 30K vocabulary
  - Interpolation with a big LM built from many other sources
- Results
  - Follow the same trend as perplexity
  - Match best published results on the same setup
  - WER reduction significant at 0.001

| Model | w/o Interp. | w/ Interp. |
|---|---|---|
| Kneser-Ney | 14.4 | 13.5 |
| CLM | 14.2 | 13.4 |
| SCLM | 13.7 | 13.3 |
| Hier. Pitman-Yor | 14.1 | 13.4 |
| CHPYLM | 13.7 | 13.2 |
| SCHPYLM | **13.5** | **13.1** |

Table: Lattice-rescoring WERs on IBM RT-04 CTS

## Conclusions

- We proposed
  - Soft-clustering Class-based Hierarchical Pitman-Yor Language Models
  - A simple collapsed Gibbs sampler for inference
- Great performance in perplexity and word error rate
  - 22% perplexity reduction on WSJ
  - 6% WER reduction on IBM RT-04 CTS
- Many thanks to Max Bisani, Ernie Pusateri, Wilson Tam and Paul Vozila for stimulating discussions.