Hierarchical Pitman-Yor Language Models for ASR in Meetings

Songfang Huang and Steve Renals

The Centre for Speech Technology Research, School of Informatics, University of Edinburgh

{s.f.huang,s.renals}@ed.ac.uk

Introduction

- The hierarchical Pitman-Yor language model (HPYLM) [Teh, ACL'06] provides a Bayesian interpretation of language model (LM) smoothing.
- An approximation to the HPYLM recovers the exact formulation of the interpolated Kneser-Ney (IKN) smoothing method.
- We focus on the application and scalability the HPYLM.
- We verify HPYLM on a large-vocabulary automatic speech recognition (LVASR) system for multiparty meeting transcription.

Pitman-Yor Process

Pitman-Yor process $\mathrm{PY}(d,\theta,G_0)$ is a generalization of Dirichlet process.

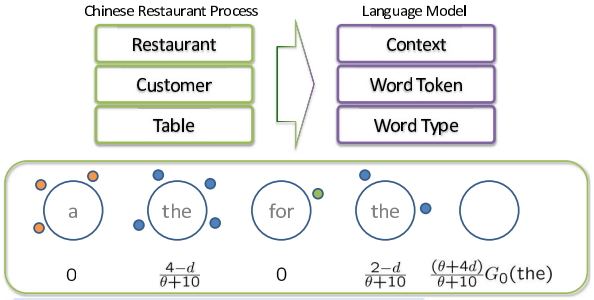
- G_0 is a base distribution, a putative mean of draws from $\mathrm{PY}(d,\theta,G_0)$.
- θ is a strength parameter, with d controls the variabilities around G_0.
- d is a discount parameter, when $d=0$ $\mathrm{PY}(0,\theta,G_0)$ reverts to $\mathrm{DP}(\theta,G_0)$.

Chinese Restaurant Metaphor for PY

A restaurant with infinite number of tables, each with infinite capacity:

1st customer sits at 1st table, next customer at: $\begin{cases}\frac{c_k-d}{\theta+c_\cdot} & \text{an occupied table}\\ \frac{\theta+dt}{\theta+c_\cdot} & \text{a new table}\end{cases}$

Pitman-Yor Process for Unigram LMs



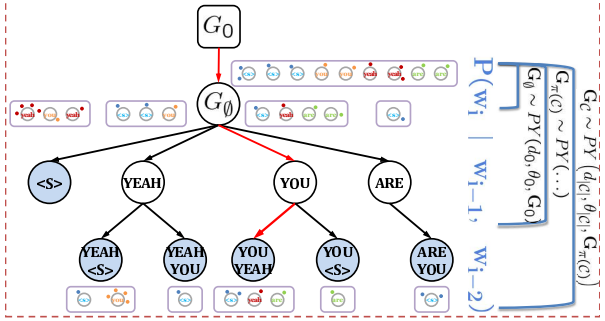
The probability of next customer "the" sitting at a specific table.

$$P(w_i=\text{"the"}|\text{seating arrangement})=\frac{c_w-dt_w}{\theta+c_\cdot}+\frac{\theta+dt_\cdot}{\theta+c_\cdot}G_0(w)$$

- PY produces a power-law distributions over the number of customers
- PY resembles the heavy-tailed distributions in natural languages.
- PY is among the family of priors for the non-parametric models.

Hierarchical Pitman-Yor LMs

Hierarchy



Training – Gibbs Sampling

- **RemoveCustomer (u, w)**
  - $\propto c_{uwk}$: remove a customer from k[th] table in u
  - if k[th] table is empty, **RemoveCustomer** (π(u), w)
- **AddCustomer (u, w)**
  - $\max(0, c_{uwk}-d_{|\mathbf{u}|})$: sit customer at k[th] table in u
  - $(\theta_{|\mathbf{u}|}+d_{|\mathbf{u}|}t_{|\mathbf{u}|})*P(\pi(\mathbf{u}),w)$: sit customer at a new table labelled w in u, **AddCustomer** (π(u), w)

Testing

- Approximate the integral by averaging over I samples from posteriors

$$P(w|\mathbf{u})=\int P(w|\mathbf{u},\mathcal{S},\Phi)P(\mathcal{S},\Phi|\mathcal{D})d(\mathcal{S},\Phi)\approx\sum_{i=1}^{I}P(w|\mathbf{u},\mathcal{S}^i,\Phi^i)/I$$

Equivalence to Interpolated Kneser-Ney

$$P^{\mathrm{IKN}}(w|\mathbf{u})=\underbrace{\frac{c_{\mathbf{u}w}-d_{|\mathbf{u}|}}{c_{\mathbf{u}\cdot}}}_{\text{Discount}}+\underbrace{\frac{d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}}{c_{\mathbf{u}\cdot}}}_{\text{Interpolation}}\underbrace{P^{\mathrm{IKN}}(w|\pi(\mathbf{u}))}_{\text{Modified Counts}}$$

$$P^{\mathrm{HPY}}_{\mathcal{S},\Phi}(w|\mathbf{u})=\frac{c_{\mathbf{u}w}-d_{|\mathbf{u}|}t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|}+c_{\mathbf{u}\cdot}}+\frac{\theta_{|\mathbf{u}|}+d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}}{\theta_{|\mathbf{u}|}+c_{\mathbf{u}\cdot}}P^{\mathrm{HPY}}_{\mathcal{S},\Phi}(w|\pi(\mathbf{u}))$$

$d_{|\mathbf{u}|}t_{\mathbf{u}w}$ in PY; Tree Hierarchy; $c_{\mathbf{u}w}=\sum_{\{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}\}}t_{\mathbf{u}'w}$

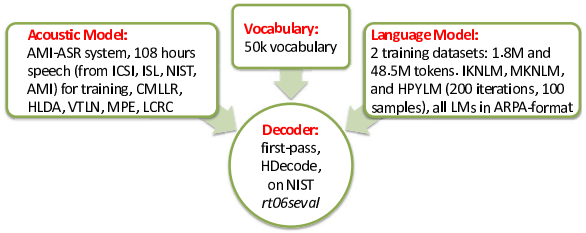
Implementation

- Implemented as an extended tool based on the SRILM toolkit
- Efficient data structures and inherited classes from SRILM
- Flexible to use with other standard smoothing methods
- Extensible for future developments, i.e., with LDA
- A standard ARPA-format LM output

Experiments and Results

Experimental Setup

We tested HPYLM in a LVASR system for meeting transcription on NIST *rt06eval*



PPL and WER Results

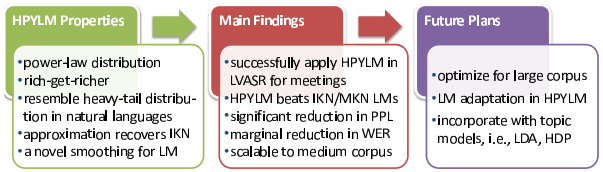
TRAIN DATA	LM Models	PERPLEXITY	WORD ERROR RATE			
			SUB	DEL	INS	TOT
Dataset-1 (1.8M tokens)	IKNLM	110.1	15.7	9.9	2.9	28.5
	MKNLM	106.5	15.6	10.0	2.8	28.4
	HPYLM	**101.2**	**15.3**	**10.1**	**2.7**	**28.1**
Dataset-2 (48.5M tokens)	IKNLM	102.9	14.6	10.0	2.6	27.3
	MKNLM	100.8	14.6	9.9	2.5	27.0
	HPYLM	**95.1**	**14.4**	**10.0**	**2.6**	**26.9**

Time and Memory for HPYLM

TRAIN DATA	WORD TOKENS	VOCABULARY	TIME/ITERATION	MEMORY
Dataset-1	**1.8 M**	50k	~10 sec	~150 MB
Dataset-2	**48.5M**	50k	~600 sec	~2400 MB*

**For IKNLM and MKNLM trained on Dataset-2, the memory requirement is around 1GB.*

Summary



This work is jointly supported by the Wolfson Microelectronics Scholarship and the European IST Programme Project FP6-033812 AMIDA.