# $p(\text{conclusions}|\text{Skipping } \{*2*\})$

Bayesian Language Modelling with Skipgrams

LANGUAGE MACHINES

CLS | Centre for Language Studies
Radboud University

# Bayesian Language Modelling with Skipgrams

Louis Onrust
Centre for Language Studies, Radboud University
Center for Processing Speech and Images, KU Leuven

l.onrust@let.ru.nl
github.com/naiaden

# Scope of the Project

**Scope**

- Language models
- Latent variable models
- Domain-dependence of LVLM
- Intrinsic & extrinsic evaluation

**Goal**

- Bring back language modelling in Bayesian language modelling
- Improve cross domain langauge modelling with skipgrams

# Language Model

**Traditional method**

The process:

- Read $n$-gram $p$
- Increment frequency of $p$
- Repeat, preferably ad infinitum

$n$-gram probabilities are then determined by their MLE

**Smoothed Traditional Language Model**

What to do when the occurrence count of $p$ is 0?

- Not assign 0 as probability $\rightarrow$ smoothing
- Fall back to the last $(n-1)$ words of $p$ $\rightarrow$ backoff

One of the best methods is still Modified Kneser-Ney: backoff and smoothing

# Language Model

**Traditional method**

The process:

- Read $n$-gram $p$
- Increment frequency of $p$
- Repeat, preferably ad infinitum

$n$-gram probabilities are then determined by their MLE

**Smoothed Traditional Language Model**

What to do when the occurrence count of $p$ is 0?

- Not assign 0 as probability $\rightarrow$ smoothing
- Fall back to the last $(n-1)$ words of $p$ $\rightarrow$ backoff

One of the best methods is still Modified Kneser-Ney: backoff and smoothing

# Language Model

**Bayesian method**

- Assume texts are generated by some process
- Consider the texts to be a sample from the process
- Infer underlying process

**Bayesian Unigram Language Model: Chinese Restaurant Process**

- Clusters are tables, unigram tokens are customers
- Initially tokens seat at the same type table
- In the inference step, customers get to choose a new identity

**Bayesian *n*-gram Language Model: Nested Chinese Restaurant Process**

- Each context is a restaurant
- Each $n$ is a floor
- Each $n$-gram is a table
- Each $(n-1)$-gram sits at a table on the $(n-1)$th floor
- All restaurants share the same global menu

# Language Model

**Bayesian method**

- Assume texts are generated by some process
- Consider the texts to be a sample from the process
- Infer underlying process

**Bayesian Unigram Language Model: Chinese Restaurant Process**

- Clusters are tables, unigram tokens are customers
- Initially tokens seat at the same type table
- In the inference step, customers get to choose a new identity

**Bayesian $n$-gram Language Model: Nested Chinese Restaurant Process**

- Each context is a restaurant
- Each $n$ is a floor
- Each $n$-gram is a table
- Each $(n-1)$-gram sits at a table on the $(n-1)$th floor
- All restaurants share the same global menu

# Language Model

**Bayesian method**

- Assume texts are generated by some process
- Consider the texts to be a sample from the process
- Infer underlying process

**Bayesian Unigram Language Model: Chinese Restaurant Process**

- Clusters are tables, unigram tokens are customers
- Initially tokens seat at the same type table
- In the inference step, customers get to choose a new identity

**Bayesian $n$-gram Language Model: Nested Chinese Restaurant Process**

- Each context is a restaurant
- Each $n$ is a floor
- Each $n$-gram is a table
- Each $(n-1)$-gram sits at a table on the $(n-1)$th floor
- All restaurants share the same global menu

# Bayesian Language Model: Learning & Estimating

**Chinese Restaurant Process: Empirical Distribution**

- Each $n$-gram enters the restaurant, and goes to the $n$th floor, to the room that represents the context
- There he seeks for the table with other $n$-grams of the same type
  - If there is such a table, he joins that table
  - Otherwise he seats himself at an empty table
- For each new table, a family member of the same $n$-gram but of length $(n-1)$ is sent to represent the family
  - This process repeats for $0 < x \leq n$

**Chinese Restaurant Process: Inference**

With $m$ customers in the restaurant, a customer re-enters the restaurant and sits a table $t$ with probability

- $\frac{1}{m+1}$ with another $n$-gram $p$, or $\frac{|t|}{m+1}$ at the same table as $p$
- $\frac{1}{m+1}$ at a new table

The number of tables grows logarithmically

# Bayesian Language Model: Learning & Estimating

**Chinese Restaurant Process: Empirical Distribution**

- Each $n$-gram enters the restaurant, and goes to the $n$th floor, to the room that represents the context
- There he seeks for the table with other $n$-grams of the same type
  - If there is such a table, he joins that table
  - Otherwise he seats himself at an empty table
- For each new table, a family member of the same $n$-gram but of length $(n-1)$ is sent to represent the family
  - This process repeats for $0 < x \leq n$

**Chinese Restaurant Process: Inference**

With $m$ customers in the restaurant, a customer re-enters the restaurant and sits a table $t$ with probability

- $\frac{1}{m+1}$ with another $n$-gram $p$, or $\frac{|t|}{m+1}$ at the same table as $p$
- $\frac{1}{m+1}$ at a new table

The number of tables grows logarithmically

# Processes and Priors

### The Generative Model

We described a Chinese restaurant process mixture model

$$\pi_{[M]} \sim \text{CRP}(M) \tag{1}$$

$$\phi_t | \pi_{[M]} \sim G_0 \qquad \text{for } t \in \pi_{[M]}, \tag{2}$$

$$x_i | \phi, \pi_{[M]} \sim F(\phi_t) \qquad \text{for } t \in \pi_{[M]} \text{ and } i \in t \tag{3}$$

### Nested Pitman-Yor Chinese Restaurant Process

- CRP and DPCRP give logarithmic growth
- Language manifests typically in power law growth
- PYCRP as generalisation of CRP and DPCRP

| | |
|---|---|
| **CRP** | No parameters |
| **DPCRP** | Concentration parameter $\alpha$ |
| **PYCRP** | Concentration parameter $\alpha$ and discount parameter $\gamma$ |

# Processes and Priors

### The Generative Model

We described a Chinese restaurant process mixture model

$$\pi_{[M]} \sim \mathrm{CRP}(M) \tag{1}$$

$$\phi_t | \pi_{[M]} \sim G_0 \qquad \text{for } t \in \pi_{[M]}, \tag{2}$$

$$x_i | \phi, \pi_{[M]} \sim F(\phi_t) \qquad \text{for } t \in \pi_{[M]} \text{ and } i \in t \tag{3}$$

### Nested Pitman-Yor Chinese Restaurant Process

- CRP and DPCRP give logarithmic growth
- Language manifests typically in power law growth
- PYCRP as generalisation of CRP and DPCRP

| | |
|---|---|
| **CRP** | No parameters |
| **DPCRP** | Concentration parameter $\alpha$ |
| **PYCRP** | Concentration parameter $\alpha$ and discount parameter $\gamma$ |

# Processes and Priors

**A Suboptimal Unigram Language Model**

We described a Chinese restaurant process mixture model

$$G_0 = \mathcal{U} \tag{1}$$
$$G \sim \text{CRP}(G_0) \tag{2}$$
$$x_i \sim G \tag{3}$$

**Nested Pitman-Yor Chinese Restaurant Process Mixture Model**

$$G_0 = \mathcal{U} \tag{4}$$
$$G_1 \sim \text{PYCRP}(\alpha_1, \gamma_1, G_0) \tag{5}$$
$$G_u \sim \text{PYCRP}(\alpha_{|u|}, \gamma_{|u|}, G_{\pi(u)}) \tag{6}$$
$$x_i | u_j \sim G_{u_j} \tag{7}$$

# Bayesian Language Model: The Implementation

**Motivation**

Existing Bayesian language models...

- are merely an algorithmic showcase without real language modelling aspirations
- cannot handle really big data sets

**Implementation**

We use the following software:

| | |
|---|---|
| **cpyp** | an existing C++ framework on BNP with PYP priors |
| **colibri** | an existing C++ framework for pattern modelling |

**Advantages**

- We can now handle many patterns such as *n*-grams, skipgrams, and flexgrams
- Tresholding patterns on many levels

# Bayesian Language Model: The Implementation

**Motivation**

Existing Bayesian language models...

- are merely an algorithmic showcase without real language modelling aspirations
- cannot handle really big data sets

**Implementation**

We use the following software:

| | |
|---|---|
| **cpyp** | an existing C++ framework on BNP with PYP priors |
| **colibri** | an existing C++ framework for pattern modelling |

**Advantages**

- We can now handle many patterns such as *n*-grams, skipgrams, and flexgrams
- Tresholding patterns on many levels

# Results: The Setup

**Data Sets**

- JRC-Acquis English
- Google 1 billion words
- EMEA English
- Wikipedia English

**Backoff Methods**

| | |
|---|---|
| **ngram** | full recursive backoff to shorter $n$-grams |
| **limited** | recursive backoff to all patterns $\leq n$ until match |
| **full** | recursive backoff to all patterns $\leq n$ |

**Evaluation Measure**

- Intrinsic evaluation with perplexity

# Results: The Setup

**Data Sets**

- JRC-Acquis English
- Google 1 billion words
- EMEA English
- Wikipedia English

**Backoff Methods**

| | |
|---|---|
| **ngram** | full recursive backoff to shorter $n$-grams |
| **limited** | recursive backoff to all patterns $\leq n$ until match |
| **full** | recursive backoff to all patterns $\leq n$ |

**Evaluation Measure**

- Intrinsic evaluation with perplexity

# Results: The Setup

**Data Sets**

- JRC-Acquis English
- Google 1 billion words
- EMEA English
- Wikipedia English

**Backoff Methods**

| | |
|---|---|
| **ngram** | full recursive backoff to shorter *n*-grams |
| **limited** | recursive backoff to all patterns $\leq n$ until match |
| **full** | recursive backoff to all patterns $\leq n$ |

**Evaluation Measure**

- Intrinsic evaluation with perplexity

# Results: An Overview

**Summary**

- Within-domain evaluation yields best performance
- Adding skipgrams increases performance on cross-domain evaluation
- For generic corpora, limited recursive backoff performs best
- Seems to outperform Generalised Language Model
- If significant, perhaps not enough for extrinsic evaluation

# Results: Domains and Patterns

## Observations

**domains** Within-domain evaluation yields best performance

**patterns** Adding skipgrams increases performance on cross-domain evaluation

| | *n*-gram | | | | skipgram | | | |
|---|---|---|---|---|---|---|---|---|
| | jrc | 1bw | emea | wp | jrc | 1bw | emea | wp |
| jrc | 13 | 1195 | 961 | 1011 | 13 | 1162 | 939 | 1008 |
| 1bws | 768 | 158 | 945 | 493 | 751 | 162 | 921 | 507 |
| emea | 600 | 1143 | 4 | 843 | 581 | 1155 | 4 | 842 |
| wps | 555 | 455 | 1005 | 217 | 565 | 470 | 990 | 227 |

# Results: Effect of Different Backoff Methods

## Observations

**backoff** For generic corpora, limited recursive backoff performs best

| | | *n*-gram | | | | skipgram | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | jrc | 1bw | emea | wp | jrc | 1bw | emea | wp |
| jrc | ngram | 13 | 1510 | 1081 | 1293 | 13 | 1843 | 1295 | 1623 |
| | limited | 14 | 1477 | 1122 | 1263 | 13 | 1542 | 1149 | 1356 |
| | full | 69 | 1195 | 961 | 1011 | 65 | **1162** | **939** | **1008** |
| 1bws | ngram | 768 | **158** | 946 | **493** | 879 | 163 | 1105 | 550 |
| | limited | 815 | 185 | 1025 | 563 | **751** | 162 | **921** | 507 |
| | full | 800 | 264 | 1039 | 583 | 769 | 252 | 988 | 561 |
| emea | ngram | 769 | 1552 | 4 | 1097 | 969 | 2090 | 4 | 1416 |
| | limited | 779 | 1385 | 4 | 1018 | 838 | 1655 | 4 | 1139 |
| | full | 600 | **1143** | 32 | 843 | **581** | 1155 | 32 | **842** |
| wps | ngram | **555** | **455** | 1005 | **217** | 623 | 504 | 1,132 | 233 |
| | limited | 629 | 543 | 1168 | 260 | 565 | 470 | **990** | 227 |
| | full | 656 | 579 | 1184 | 357 | 625 | 548 | 1,106 | 336 |

# Future Work

**Experiments**

- Validate significance by testing on multiple languages
- Investigate influence skipgrams with qualitative analysis
- When we find a more substantial drop in perplexity:
  - Machine translation experiments
  - Automated speech recognition experiments
- Investigate multi-domain language models (DHPYPLM)
- Generalise skipgrams to flexgrams
- ...