Compressed Nonparametric Language Modelling

Ehsan Shareghi  
Monash University

Gholamreza Haffari  
Monash University

Trevor Cohn  
The University of Melbourne

IJCAI 2017
Outline

• Infinite-Order Language Modelling and Challenges
• Compressed HPYP LM
• Inference and Sampling in Compressed HPYP LM
• Perplexity and Mixing
• Conclusion and Future Directions
Language Modelling (LM)

Predictive typing/Auto completion

Machine Translation

Donald trump is a p
- donald trump is a pokemon
- donald trump is a potato
- donald trump is a populist
- donald trump is a politician
- donald trump is a pragmatist
- donald trump is a prophet
- donald trump is a piece of garbage
- donald trump is a pendejo

Président de la Chambre des représentants

President of the Bedroom of Representatives
President of the House of Representatives

P(House | President of the) > P(Bedroom | President of the)
Infinite order LM

\[ P(w_1^N) = \prod_{i=1}^{N} P(w_i | w_{1}^{i-1}) \]

Statistical sparsity
Solution: smoothing (HPYP, etc)

Computational cost of smoothing
No Scalable Solution
Why Infinite-order LM?

Data Size vs Perplexity:
- Finite (n=10) and Infinite models are compared.
- As data size increases from 125MiB to 8GiB, the perplexity decreases for both models.
- The Infinite models show a more consistent decrease in perplexity compared to the Finite models.

<table>
<thead>
<tr>
<th>DATA SIZE</th>
<th>Finite (n=10)</th>
<th>Infinite</th>
</tr>
</thead>
<tbody>
<tr>
<td>125MiB</td>
<td>350</td>
<td>290</td>
</tr>
<tr>
<td>250MiB</td>
<td>330</td>
<td>270</td>
</tr>
<tr>
<td>1GiB</td>
<td>310</td>
<td>250</td>
</tr>
<tr>
<td>2GiB</td>
<td>290</td>
<td>230</td>
</tr>
<tr>
<td>4GiB</td>
<td>270</td>
<td>210</td>
</tr>
<tr>
<td>8GiB</td>
<td>250</td>
<td>190</td>
</tr>
</tbody>
</table>
Computational Cost of HPYP LM - Training

Involved Factors:
- Building Model (hierarchy)
- Parameters Sampling
- Storing the Model and Parameters
Outline

• Infinite-Order Language Modelling and Challenges
• Compressed HPYP LM
• Inference and Sampling in Compressed HPYP LM
• Perplexity and Mixing
• Conclusion and Future Directions
Hierarchical Pitman-Yor Process (HPYP)

\[ \tilde{G}_u \sim \text{PYP}(\theta_u, d_u, \tilde{G}_{\pi(u)}) \]
\[ \tilde{G}_\varepsilon \sim \text{PYP}(\theta_\varepsilon, d_\varepsilon, \frac{1}{|\text{vocab}|}) \]

Same model as “Sequence Memoizer” Wood et al. (2011)
\( u = \text{“from rich”}, \quad \eta_u = \{d_u, \theta_u, \{n_u^w, t_u^w\}_{w \in u}\} \)

\( n_u^{\text{give}} = 6, \quad t_u^{\text{give}} = 2 \)
HPYP LM – Chinese Restaurant Process

\( u = \text{“from rich”}, \quad \eta_u = \{d_u, \theta_u, \{n_u^w, t_u^w\}_{w \in u}\} \)

\[ n_u^w = 6, \quad t_u^\text{give} = 2 \]

\[ 0 \leq t_u^w \leq n_u^w \]

\[ n_{\pi(u)}^w = \sum_{v \in \text{children} (\pi(u))} t_v^w \]
Compressed HPYP LM

- Hierarchy of KN and HPYP LMs are the same
- KN can serve as an approximate inference for HPYP ($\theta_u = 0$ and $t_u^w = 1$)
Compressed HPYP LM

- Hierarchy of KN and HPYP LMs are the same
- KN can serve as an approximate inference for HPYP ($\theta_u = 0$ and $t_u^w = 1$)

- KN hierarchy can be recovered from a **compressed suffix tree** of data on-the-fly
- **Compressed Suffix Trees:**
  - Based on advanced data structures such as the Wavelet Tree of the BWT of text
  - Contain all the information about the HPYP hierarchy and text itself in a space matching the text size

---

Shareghi et al., 2016, *Fast, Small and Exact Language Modelling with Compressed Suffix Trees, Transactions of the ACL*
Compressed HPYP LM

- Hierarchy of KN and HPYP LMs are the same
- KN can serve as an approximate inference for HPYP ($\theta_u = 0$ and $t_u^w = 1$)

- KN hierarchy can be recovered from a compressed suffix tree of data on-the-fly
- Compressed Suffix Trees:
  - Based on advanced data structures such as the Wavelet Tree of the BWT of text
  - Contain all the information about the HPYP hierarchy and text itself in a space matching the text size

<table>
<thead>
<tr>
<th>Compressed HPYP</th>
<th>HPYP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs Compressed Suffix Tree of Data</td>
<td>Constructs Hierarchy of HPYP</td>
</tr>
<tr>
<td>No Sampling</td>
<td>Samples across all nodes and for all $w$</td>
</tr>
</tbody>
</table>
Training time comparison

- **MEMORY [GiB]**
  - 125MIB
  - 250MIB
  - 1GIB
  - 2GIB
  - 4GIB
  - 8GIB

- **DATA SIZE**
  - 1+ GiB
  - 91+ GiB

- **TIME [MINUTES]**
  - 1+ hour
  - 20+ days

- **DATA SIZE**
  - 125MIB
  - 250MIB
  - 1GIB
  - 2GIB
  - 4GIB
  - 8GIB
Outline

- Infinite-Order Language Modelling and Challenges
- Compressed HPYP LM
- Inference and Sampling in Compressed HPYP LM
- Perplexity and Mixing
- Conclusion and Future Directions
Inference in Compressed HPYP

We need to compute the following \textit{intractable} integral,

\[
P(w|u) = \int P(w|u, \eta) P(\eta) \, d\eta
\]

which we approximate using samples for \(\eta_u = \{d_u, \theta_u, \{n^w_u, t^w_u\}_{w \in u}\}\).

\[
0 \leq t^w_u \leq n^w_u
\]

\[
n^w_{\pi(u)} = \sum_{v \in \text{children}(\pi(u))} t^w_v
\]
Sampling $\{n_u^w, t_u^w\}_{w \in u}$

Given a query $P(\text{give} \mid \text{from rich})$
Sampling \( \{ n_w^{u}, t_w^{u} \}_{w \in u} \)

Given a query \( P(\text{give} | \text{from rich}) \)
Sampling $\{n_w^u, t_w^u\}_{w \in \mathcal{U}}$

Given a query $P(\text{give} \mid \text{from rich})$
Sampling \( \{ n\^w_u, t\^w_u \}_{w \in u} \)

Given a query \( P(\text{give} \mid \text{from rich}) \)

Read \( n \) give \( \text{from rich} \) from Data

KN Initialized
Sampling $\{n^w_u, t^w_u\}_{w \in u}$

Given a query $P(\text{give} \mid \text{from rich})$

Sample $t^\text{give}_{\text{from rich}} = 2$

$P(give \mid from\ rich)$
Sampling \( \{ n^w_u, t^w_u \}_{w \in u} \)

Given a query \( P(\text{give} \mid \text{from rich}) \)

Sample \( t^{\text{give}} \)

from rich = 2

\( \text{Sample } t^{\text{give}} \text{ from rich } = 2 \)
Sampling $\{n_u^w, t_u^w\}_{w \in u}$

Given a query $P(\text{give} \mid \text{from rich})$

Update $n_{\text{give rich}}$

$P(give \mid from rich)$
Given a query $P(\text{give} \mid \text{from rich})$
Sampling $\{n^w_u, t^w_u\}_{w \in u}$

Given a query $P(\text{give} | \text{from rich})$
Sampling \( \{ n_u^w, t_u^w \}_{w \in u} \)

Given a query \( P(\text{give} \mid \text{from rich}) \)

Read \( n_{\text{give rich}} \) from proxy counts

KN Initialized
Given a query $P(\text{give} \mid \text{from rich})$

Read $n_{\text{give}}$ from proxy counts

and so on
Sampling $\{n^w_u, t^w_u\}_{w \in u}$

Given a query $P(\text{give} \mid \text{from rich})$

Read $n^\text{give}_u$ from proxy counts

and so on

In practice:
- $0 \leq t^w_u \leq \text{Min}(n^w_u, M)$
- Generate 100 samples per node
- Forget samples
Test time comparison

On 1 GiB:
- 1.8x slower on query
- 2.3x faster on load+query
Outline

• Infinite-Order Language Modelling and Challenges
• Compressed HPYP LM
• Inference and Sampling in Compressed HPYP LM
• Perplexity and Mixing
• Conclusion and Future Directions
Perplexity (and Mixing)

![Bar chart showing perplexity for different data sizes with three methods: KN (n=10), Full HPYP, and Our Approach.]

- **DATA SIZE**
  - 125MiB
  - 250MiB
  - 1GiB
  - 2GiB
  - 4GiB
  - 8GiB

- **Perplexity**
  - 0
  - 50
  - 100
  - 150
  - 200
  - 250
  - 300
  - 350

- **Methods**
  - KN (n=10)
  - Full HPYP
  - Our Approach
Outline

- Infinite-Order Language Modelling and Challenges
- Compressed HPYP LM
- Inference and Sampling in Compressed HPYP LM
- Perplexity and Mixing
- Conclusion and Future Directions
Conclusion

- Proposed a Compressed HPYP LM and a fast and memory-efficient approximate inference scheme.
- Proposed approach is several orders of magnitude smaller than the existing models.
- Avoided potential mixing issues, while consistently outperforming the state-of-the-art count-based language models by a significant margin.
Conclusion

• Proposed a Compressed HPYP LM and a fast and memory-efficient approximate inference scheme.
• Proposed approach is several orders of magnitude smaller than the existing models.
• Avoided potential mixing issues, while consistently outperforming the state-of-the-art count-based language models by a significant margin.

Future Directions

• Sampling speedup (i.e., learning an approximation for Stirling numbers)
• Exploring continuous space approximations of HPYP
• Exploring other applications
Conclusion

- Proposed a Compressed HPYP LM and a fast and memory-efficient approximate inference scheme.
- Proposed approach is several orders of magnitude smaller than the existing models.
- Avoided potential mixing issues, while consistently outperforming the state-of-the-art count-based language models by a significant margin.

Thanks!

Compressed Nonparametric Language Modelling
Slides, supplementary materials, more results available on: eehsan.github.io
Contact: Ehsan.Shareghi@gmail.com