Skipping meals, skipping words, and how the latter can benefit you and the first just makes you hungry

Bayesian Language Modelling with Skipgrams

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

Language Models

Applications

- Input assists on telephones
- Automatic translation of search results
- Digital court reporting

Language Models

Applications

- Input assists on telephones
- Automatic translation of search results
- Digital court reporting

Flavours

- Frequentist language models
- Bayesian language models
- Neural language models
- ...

The Task at Hand

After all , tomorrow is another [. . .]

The Task at Hand

After all , tomorrow is another [. . .]

Word prediction **Word probability** **Pattern probability**

The Task at Hand

After all , tomorrow is another [. . .]

Word prediction	**Word probability**	**Pattern probability**
• day		

The Task at Hand

After all , tomorrow is another [. . .]

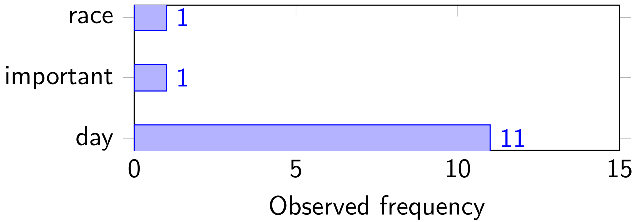
Word prediction

- day

Word probability

- cow
- day
- vegetable
- . . .

Pattern probability



The Task at Hand

After all , tomorrow is another [. . .]

Word prediction

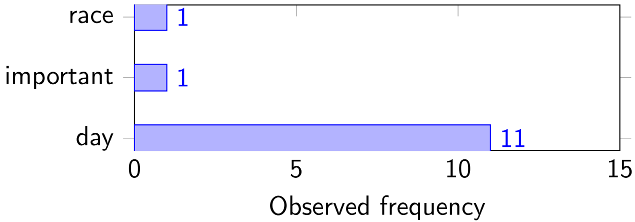
- day

Word probability

- cow
- day
- vegetable
- . . .

Pattern probability

- tomorrow is another day
- tomorrow is another important
- tomorrow is another race



Generalising the *n*-gram

***n*-grams**

- Continuous sequence of *n* words

Skipgrams

- *n*-gram with at most $n - 2$ skips of length 1

Flexgrams

- *n*-gram with any number of skips of any length

Skipgrams in other works

✎ A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser Ney Smoothing, Pickhardt et alia, 2014

✎ Skip-gram Language Modeling Using Sparse Non-negative Matrix Probability Estimation, Shazeer et alia, 2014

✗ Skipgrams in word2vec

Backoff Patterns with Skipgrams

tomorrow is another day

is another day   tomorrow {1} another day   tomorrow is {1} day

another day   is {1} day   tomorrow {2} day

day

∅

Backoff patterns

- **simple** Only n-grams
- **limited** All patterns until known pattern
- **full** All patterns until words

Probability Estimation

Maximum Likelihood Estimate

$$p_{\mathrm{ML}}(w_i | w_{i-N+1}, \ldots, w_{i-1}) = \frac{C(w_{i-N+1}, \ldots, w_i)}{C(w_{i-N+1}, \ldots, w_{i-1})}$$

- Parameter estimation is impossible for $N > 2$
- Naïve priors assuming independent parameters fail as well

Probability Estimation

Maximum Likelihood Estimate

$$p_{\mathrm{ML}}(w_i|w_{i-N+1},\ldots,w_{i-1}) = \frac{C(w_{i-N+1},\ldots,w_i)}{C(w_{i-N+1},\ldots,w_{i-1})}$$

- Parameter estimation is impossible for $N > 2$
- Naïve priors assuming independent parameters fail as well

Smoothing

$$p_{\mathrm{SM}}(w_i|w_{i-N+1},\ldots,w_{i-1}) = \sum_{n=1}^{N} \lambda(n) Q_n(w_i|w_{i-N+1},\ldots,w_{i-1})$$

- Chen and Goodman found that interpolated and modified Kneser-Ney are best under virtually all circumstances

Bayesian Probability Estimation

Parametrise conditional probabilities

$$p(w_i = w | w_{i-N+1}, \ldots, w_{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in W}$$

$$\pi(w_{i-N+1}, \ldots, w_{i-1}) = w_{i-N+2}, \ldots, w_{i-1}$$

- G_u is a probability vector associated with context u

Bayesian Probability Estimation

Parametrise conditional probabilities

$$p(w_i = w | w_{i-N+1}, \ldots, w_{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in W}$$

$$\pi(w_{i-N+1}, \ldots, w_{i-1}) = w_{i-N+2}, \ldots, w_{i-1}$$

- G_u is a probability vector associated with context u

Hierarchical Dirichlet language model

- What is $p(G_u | G_{\pi(u)})$?
- Standard Dirichlet distribution over probability vectors: does not outperform ikn and mkn (MacKay and Peto, 1994)

Bayesian Probability Estimation

Parametrise conditional probabilities

$$p(w_i = w | w_{i-N+1}, \ldots, w_{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in W}$$

$$\pi(w_{i-N+1}, \ldots, w_{i-1}) = w_{i-N+2}, \ldots, w_{i-1}$$

- G_u is a probability vector associated with context u

Hierarchical Dirichlet language model

- What is $p(G_u | G_{\pi(u)})$?
- Standard Dirichlet distribution over probability vectors: does not outperform ikn and mkn (MacKay and Peto, 1994)

Hierarchical Pitman-Yor process

- Two-parameter extension of the Dirichlet distribution
- PYP produces power-law distributions
- Outperforms ikn and mkn (Teh, 2006)

Experimental Setup

Mixed-domain data

- Google 1 Billion Words
- Wikipedia November 2013

Domain-specific data

- JRC-ACQUIS: European legislation
- European Medicines Agency documents

Sampling

1bws 10% of the words
wps 5% of the words

Tresholding on 1BW

unigrams Threshold on 100: 99561 types
n-grams Thresholds 2, 5, and 10

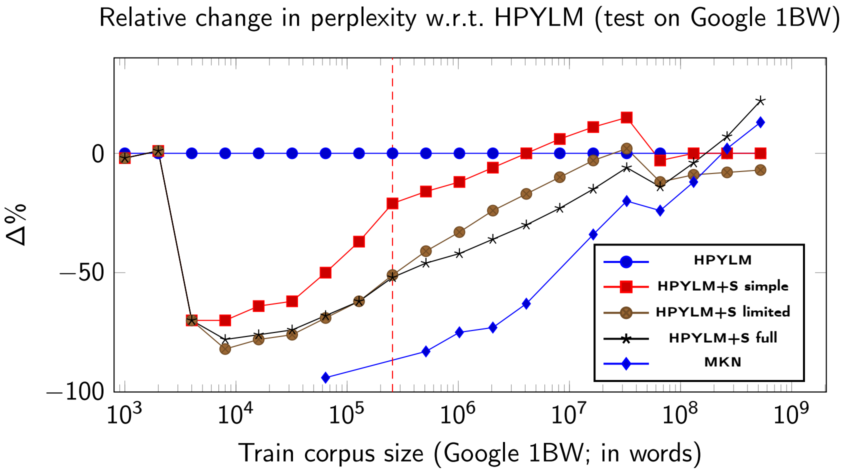
Method

cpyp C++ library for nonparametric Bayesian modelling with Pitman-Yor process priors: https://github.com/redpony/cpyp

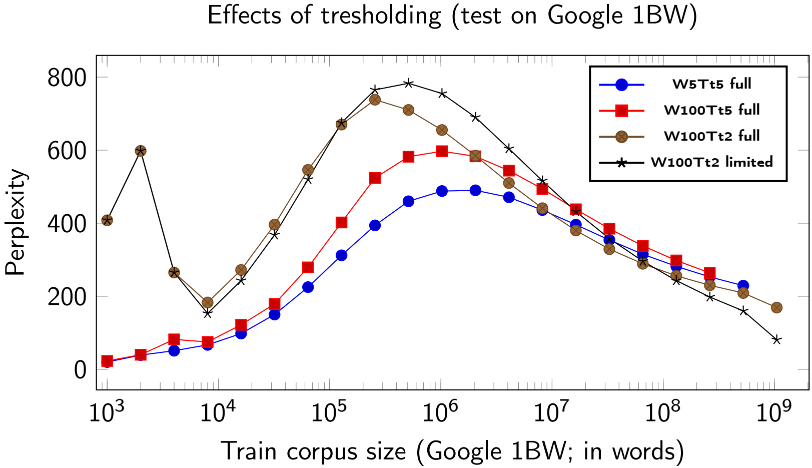
Colibri Core C++ library for working with basic linguistic constructions such as n-grams and skipgrams: https://github.com/proycon/colibri-core

cococpyp C++ toolkit for Bayesian language modelling: https://github.com/naiaden/cococpyp

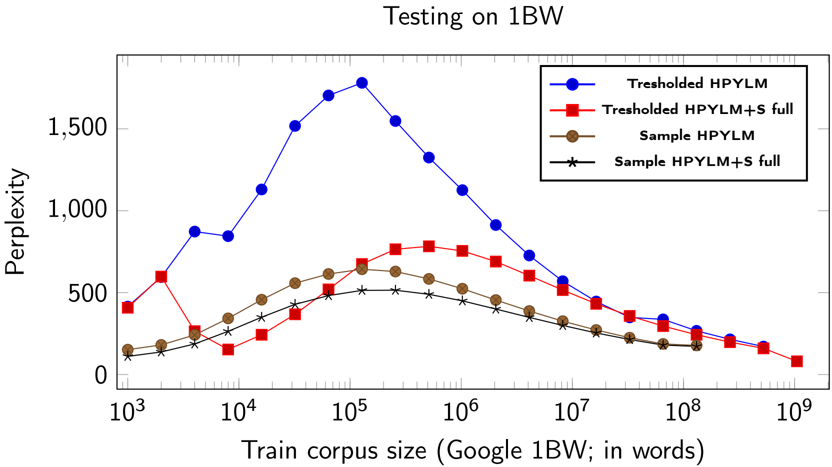
Comparing the Models



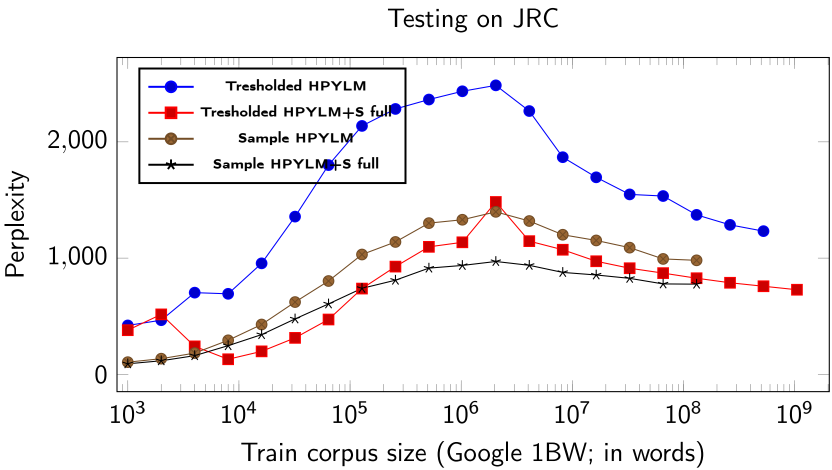
Skipping Meals



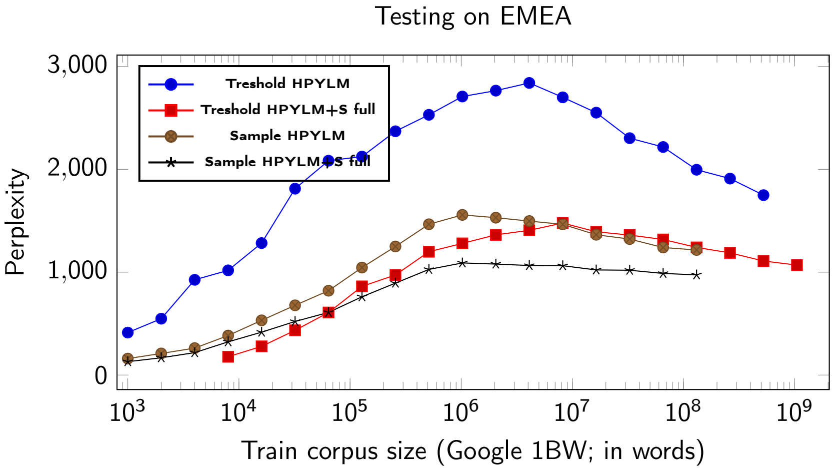
Learning Curves: Effects of Data Reduction



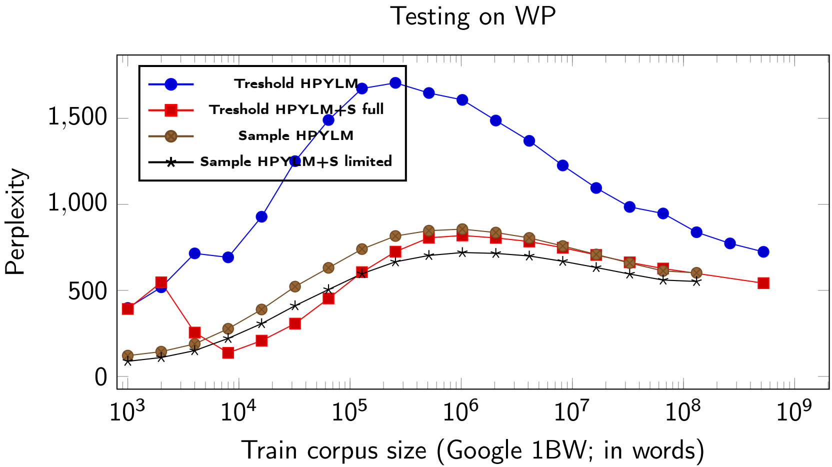
Learning Curves: Effects of Data Reduction



Learning Curves: Effects of Data Reduction



Learning Curves: Effects of Data Reduction



Contribution of Skipgrams

HPYLM

	jrc	1bw	emea	wp
jrc	13	1195	961	1011
1bw	1232	171	1749	724
1bws	768	158	946	493
emea	600	1143	4	843
wps	555	455	1005	217

HPYLM + Skipgrams

	jrc	1bw	emea	wp
jrc	13	1162	938	1008
1bw	***728***	***81***	***1069***	***542***
1bws	751	162	921	507
emea	581	1155	4	842
wps	565	470	990	227

Preliminary conclusions

Sampled versus tresholded corpus

- For in-domain evaluation we need about 2 times the amount of tresholded data
- For cross-domain evaluation we need at least 5 times as much data

When to use tresholding

- Within-domain better with tresholding, because more training data (patterns) like test data
- Cross-domain better without tresholding, because more types (100k vs. 1.1M types)

Skipgrams help in all situations

- Within-domain performance converges with ngrams
- Cross-domain performance increases (30-40% reduction in perplexity)

Intrinsic evaluation

- Apply HPYLM+S to ASR, MT, ...

Choosing a Language Model

Quick turnaround

- ✎ Improved backing-off for m-gram language modeling, Kneser & Ney, 1995

Best results

- ✎ Hierarchical Pitman-Yor process language model, Teh, 2006
- ✎ A parallel training algorithm for hierarchical Pitman-Yor process language models, Huang & Renals, 2009
- ✎ Recurrent neural network language model, Mikolov et alia, 2010

Newest

- ✎ Sparse Non-negative Matrix Language Modeling For Skip-grams, Shazeer et alia, 2014
- ✎ Language Modeling with Power Low Rank Ensembles, Parikh et alia, 2014
- ✎ Word representations via Gaussian embedding, Vilnis and MacCallum, under review