# Hierarchical Pitman-Yor Language Model for Machine Translation

Tsuyoshi Okita
*CNGL / School of Computing*
*Dublin City University*
*Dublin, Ireland*
*tokita@computing.dcu.ie*

Andy Way
*CNGL / School of Computing*
*Dublin City University*
*Dublin, Ireland*
*away@computing.dcu.ie*

***Abstract*—The hierarchical Pitman-Yor process-based smoothing method applied to language model was proposed by Goldwater and by Teh; the performance of this smoothing method is shown comparable with the modified Kneser-Ney method in terms of perplexity. Although this method was presented four years ago, there has been no paper which reports that this language model indeed improves translation quality in the context of Machine Translation (MT). This is important for the MT community since an improvement in perplexity does not always lead to an improvement in BLEU score; for example, the success of word alignment measured by Alignment Error Rate (AER) does not often lead to an improvement in BLEU. This paper reports in the context of MT that an improvement in perplexity really leads to an improvement in BLEU score. It turned out that an application of the Hierarchical Pitman-Yor Language Model (HPYLM) requires a minor change in the conventional decoding process. Additionally to this, we propose a new Pitman-Yor process-based statistical smoothing method similar to the Good-Turing method although the performance of this is inferior to HPYLM. We conducted experiments; HPYLM improved by 1.03 BLEU points absolute and 6% relative for 50k EN–JP, which was statistically significant.**

***Keywords*-Statistical Machine Translation, Statistical smoothing method, Hierarchical Pitman-Yor process.**

## I. Introduction

Statistical approaches or non-parametric Machine Learning methods estimate some targeted statistical quantities based on the (true) posterior distributions in a Bayesian manner [1] or based on the underlying fixed but unknown (joint) distributions from which we assume that we sample our training examples in a frequentist manner [17]. In NLP (Natural Language Processing), such distributions are observed by simply counting (joint/conditional) events, such as $c(w)$, $c(w_1, w_2)$ and $c(w_3|w_1, w_2)$ where $w$ denotes words and $c(\cdot)$ denotes a function to count events; since such quantities are often discrete, it is unlikely that such events will be counted incorrectly at first sight. However, it is a well-known fact in NLP that such counting methods are often unreliable if the size of the corpus is too small compared to the model complexity. Researchers in NLP often try to rectify such counting of (joint or conditional) events using a technique known as smoothing [3], [9]. Most smoothing techniques do not have a statistical model but rely on either interpolation or back-off schemes.

This paper discusses a statistical smoothing method based on (hierarchical) Pitman-Yor processes, which is a non-parametric generalization of the Dirichlet distribution that produces power-law distributions [16], [4]. Various pieces of research have been carried out in which hierarchical Pitman-Yor processes have been applied to language models (Hierarchical Pitman-Yor Language Model (HPYLM) [16], [11], [5]) whose generative model uses hierarchies of $n$-grams. This model is shown to be superior to the interpolated Kneser-Ney methods [9] and comparable to the modified Kneser Ney methods in terms of perplexity. Hierarchical Pitman-Yor processes have been successfully applied to word segmentation as well [4], [10].

This paper is organized as follows. Section 2 reviews hierarchical Pitman-Yor process and related issues. In Section 3 our algorithm is presented; 1) how to obtain HPYLM and 2) a minor change in the decoding algorithm. Experimental results are presented in Section 4. Section 5 concludes and provides avenues for further research.

## II. Pitman-Yor Process and Smoothing Methods

*Pitman-Yor Process:* The Pitman-Yor process [13] $PY(d, \theta, G_0)$ is a three-parametric distribution over a (base) distribution where $d$ is a discount parameter, $\theta$ a strength parameter, and $G_0$ a base distribution. When $d = 0$, the Pitman-Yor process reduces to a Dirichlet distribution $Dir(\theta G_0)$. This generative procedure produces a power-law distribution, indicating that many unique words are observed, most of them rarely, for the following reasons: the more words have been assigned to a draw from $G_0$, the more likely subsequent words will be assigned to the draw, while the more we draw from $G_0$, the more likely a new word will be assigned to a new draw from $G_0$.

*Chinese Restaurant Process:* Let $(x_1, \ldots, x_n)$ be a training set. When the vocabulary is finite, although $PY(d, \theta, G_0)$ has no known analytic form, we can describe the Pitman-Yor process in terms of a generative procedure that produces $x_1$,$x_2$,... iteratively by marginalized out $G$. This procedure for generating words drawn from $G$ is called a Chinese restaurant process.

Let $c_k$ be the number of words assigned the value of draw $z_k$, $c_{\cdot} = \sum_{k-1}^{t} c_k$ be the current number of draws from $G$, and $t$ be the current number of draws from $G_0$. A Chinese restaurant contains an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. The first customer sits at the first available table, while each of the subsequent customers sits at an occupied table with probability proportional to the number of customers already sitting there $c_k - d$, or at a new unoccupied table with probability proportional to $\theta + dt$. If $z_i$ is the index of the table chosen by the $i$th customer, the $i$th customer sits at table $k$ given the seating arrangement of the previous $i-1$ customers $z_i = \{z_1, \ldots, z_{i-1}\}$) with probability, as is shown in (1):

$$P(z_i = k|z_i) = \begin{cases} \frac{c_k - d}{\theta + c_{\cdot}} & 1 \leq k \leq t. \\ \frac{\theta + dt}{\theta + c_{\cdot}} & k = t+1. \end{cases} \tag{1}$$

## III. Our Algorithm

Our algorithm addresses two concerns. The first concern is to update our language model based on the hierarchical Pitman-Yor process-based smoothing method, which is described in the first subsection. The second concern is to incorporate the zero probabilities based on the hierarchical Pitman-Yor process-based smoothing method. A Phrase-Based Statistical Machine Translation (PB-SMT) decoder uses constant zero probabilities for unseen phrases, while the zero probabilities based on the language model based on the hierarchical Pitman-Yor process-based smoothing method are not constant but are different based on context, e.g. (n-1)-gram hierarchies.

### A. Language Model Smoothing

We describe the following generative model which uses the Pitman-Yor process as a prior which we use in two language models in common. For a given a context $u$, let $G_u(w)$ be the probability of the current word taking value $w$. Using a Pitman-Yor process as the prior for $G_u[G_u(w)]_{w \in W}$ as in (2):

$$G_u \sim PY(d_{|u|}, \theta_{|u|}, G_{\pi(u)}) \tag{2}$$

where $\pi(u)$ is a function whose parameter is a context $u$, the discount and strength parameters are functions of the length $|u|$ of the context, while the mean vector is $G_{\pi(u)}$, the vector of probabilities of the current word given all but the earliest word in the context.

*Hierarchical Pitman-Yor Language Model:* In the generative model which uses the Pitman-Yor process as a prior in Equation (2), let us consider $\pi(u)$ as the suffix of $u$ consisting of all but the earliest word in Equation (2) [16]. This signifies that $u$ is $n$-gram words and $\pi(u)$ is ($n$-1)-gram words; this induction of Equation (2) makes an $n$-gram hierarchy. Teh proposes a method to use the hierarchical Pitman-Yor processes recursively to place a prior over $G_{\pi(u)}$ using Equation (2), but now with parameters $d_{\pi(u)}$, $\theta_{\pi(u)}$ and mean vector $G_{\pi(\pi(u))}$ instead.

$$\begin{cases} G_u \sim PY(d_{|u|}, \theta_{|u|}, G_{\pi(u)}) \\ \ldots \\ G_{\emptyset} \sim PY(d_0, \theta_0, G_0) \end{cases} \tag{3}$$

This is repeated until we get to $G_0$, the vector of probabilities over the current word given the empty context $\emptyset$. Finally, we place a prior on $G_0$ as is shown in Equation (3), where $G_0$ is the global mean vector, given a uniform value of $G_0 = 1/V$ for all $w \in W$.

The simplest inference procedure is to build a Gibbs sampler which randomly selects $n$-gram words, draws a binary decision as to which $(n-1)$-gram words originated from, and updates the language model according to the new lower-order $n$-grams [4]. A blocked Gibbs sampler is proposed by Mochihashi et al. [10], which is originally proposed for segmentation. This algorithm is an iterative procedure, which randomly selects a $n$-gram word, removes the "sentence" data of this $n$-gram word, and updates by adding a new "sentence" according to the new $n$-grams. This procedure is expected to mix rapidly compared to the simple Gibbs sampler.

By Equation (1), the predictive distribution of $n$-gram probability in HPYLM is recursively calculated as in Equation (4):

$$p(w|h) = \frac{c(w|h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta + d \cdot t_{h\cdot}}{\theta + c(h)} p(w|h') \tag{4}$$

where $p(w|h')$ is the same probability using a ($n$-1)-gram context $h'$. The case when $t_{hw} = 1$ corresponds to an interpolated Kneser-Ney smoothing [9].[1]

*Good-Turing Pitman-Yor Language Model:* In the same generative model which uses the Pitman-Yor process as a prior in Equation (2) once (not recursively), let us now consider $\pi(u)$ as a count-counts[2] function. We refer to this model as Good-Turing Pitman-Yor Language Model (GTPYLM). It is to be noted that count-counts $n_c$, described in Equation (6), is a concept appearing in Good-Turing discounting [3]:

$$c_g = (c+1)\frac{n_{c+1}}{n_c} \tag{6}$$

where $c_g$ is a modified count value used to replace $c$ in subsequent relative frequency estimates, and $n_c$ is the number of events having count $c$. Our intention here is to make

[1]Teh explains this in this way [16]: If we restrict $t_{hw\cdot}$ to be at most 1 as in (5),

$$t_{hw\cdot} = \min(1, c_{hw\cdot}), \qquad c_{hw\cdot} = \sum_{h':\pi(h')=h} t_{h'w\cdot} \tag{5}$$

we will obtain the same discount value so long as $c_{hw\cdot} > 0$, i.e. absolute discounting. Furthermore, supposing that the strength parameters are all $\theta_{|h|} = 0$, the predictive probabilities in Equation (4) now directly reduce to the predictive probabilities given by interpolated Kneser-Ney.

[2]This is also known as event-counts or count of counts.

a prior distribution $G_u$ a power-law. Hence, this method incorporates zero-frequencies by enforcing the distribution to be a power-law.

By Equation (1), the predictive distribution of $n$-gram probability in GTPYLM is computed as in (7):

$$p(w|n_w) = \frac{c(w|n_w) - d \cdot t_{n_w w}}{\theta + c(n_w)} + \frac{\theta + d \cdot t_{n_w \cdot}}{\theta + c(n_w)} p(w|n_w - 1) \quad (7)$$

where $(n_w - 1)$ is $(n_c - 1)$ where $c = n_w$, $p(w|n_w - 1)$ denotes a $(n_w - 1)$ count-count distribution, and $p(w|n_w)$ denotes a $n_w$ count-count distribution.

### B. Decoding Algorithm in PB-SMT

A minor difference in the decoding process is required. In a test sentence, if we encounter unseen phrases, a conventional PB-SMT decoder looks up the probability with constant zero-probabilities. However, our algorithm should look up the corresponding probabilities based on the hierarchical Pitman-Yor processes. We calculate these zero-probabilities using the parameters that we derived during obtaining HPYLM.

There are two way to incorporate this: 1) just before we do decoding, we update a language model by supplying a test sentence in terms of zero-probabilities, and 2) we modify a PB-SMT decoder to incorporate this difference. Due to the easy implementation, we take the approach 1) here, but the effect would be the same.

Our procedures are follows. Firstly, we prepare HPYLM parameter file $p_0(w)$ which we obtained when we calculate HPYLM. This HPYLM parameter file contains the parameters in Chinese restaurant processes, such as the number of tables, d, $\theta$, and so forth. Such parameters enable us to calculate the zero-probabilities for unseen phrases in a test sentence. The overall algorithm to obtain updated HPYLM is shown in Algorithm 1.

**Algorithm 1** Decoder for HPYLM $p(w)$

Given: a test sentence $\breve{s} = \{\breve{s}_1, \cdots, \breve{s}_n\}$, HPYLM $p(w)$, HPYLM parameter file $p_0(w)$.

Step 1: By generating a possible $n$-gram candidate, using $p_0$ we update HPYLM $p'(w)$.

Step 2: Run a decoder which looks up updated HPYLM $p'(w)$.

## IV. Experimental Results

### A. Experimental Setup

For all the experiments, we used a standard log-linear phrase-based MT system based on Moses [8]. The GIZA++ implementation [12] of IBM Model 4 was used for word alignment, followed by the grow-diag-final heuristics as phrase extraction. We used SRILM [14] to derive a 5-gram language model. We performed MERT and use a Moses decoder. The baseline 1 derived a 5-gram language model by SRILM with modified Kneser-Ney method and the baseline 2 used with SRILM with Good-Turing method.

For the HPYLM and GTPYLM, we obtained the results by a method using a blocked Gibbs sampler [10], which was considerably more efficient compared to a conventional Gibbs sampler [4], [16]. In this experiment, we used a phrase table derived by the conventional method. Perplexity was measured in terms of the same test set.

### B. Experimental Results

We conducted an experimental evaluation for JP–EN on the NTCIR-8 corpus [2] and for FR–EN and ES–EN on Europarl [7]. We randomly extracted two training corpora of 50k and 200k sentence pairs, where we used 1,200 sentence pairs (NTCIR) and 2,000 sentence pairs (Europarl) for the development set, and 1,119 (EN-JP) / 1,251 (JP-EN) sentence pairs (NTCIR) and 2,000 sentence pairs (Europarl; test2006) for the test set.

The results are shown in Table I. HPYLM obtained the best results in all the cases; the best among them was 1.03 BLEU points absolute and 6% relative for 50k EN-JP which was statistically significant verified by bootstrap resampling [6]. GTPYLM obtained the second best results in all the cases; an improvement of 0.90 BLEU points absolute and 5% relative for 50k EN-JP. These experiments also show that the perplexity measure may be reliable for the final performance measured by BLEU score.

## V. Conclusions

This paper presents an application of the hierarchical Pitman-Yor process-based language model to MT. Firstly, although the performance of HPYLM was reported in terms of perplexity, there have been no reports, as far as we know, in terms of BLEU in the MT context. We showed that there was a gain with a minor change in the decoding process. Although Teh reported that HPYLM showed a comparable performance with the modified Kneser-Ney method, we obtained better results than the modified Kneser-Ney method here. Secondly, we proposed an alternative language model using the Pitman-Yor process applying the count-counts distribution of the Good-Turing method. The performance of this was not as successful as HPYLM, but it was better than both the modified Kneser-Ney and Good-Turing methods. Furthermore, this was statistically significant.

There are several avenues for further research. Firstly, our results for our three language pairs under 200k sentence pairs would support the basic effectiveness of this statistical smoothing method for language modelling. We would like to extend our work to different language pairs and larger data sets. Note that for the giga-sized data, this method will not be required since smoothing is a method to resolve the sparse data problem. Secondly, our experiments are limited

| size | system | EN-JP | Perplexity | JP-EN | Perplexity |
|---|---|---|---|---|---|
| 50k | baseline1 | 16.33 | 71.468 | 22.01 | 131.438 |
| 50k | baseline2 | 16.20 | 72.435 | 21.81 | 136.812 |
| 50k | HPYLM | 17.36 | 66.012 | 22.81 | 116.074 |
| 50k | GTPYLM | 17.23 | 67.112 | 22.70 | 120.320 |
| 200k | baseline1 | 23.42 | 59.607 | 21.68 | 117.78 |
| 200k | baseline2 | 23.36 | 58.587 | 21.38 | 119.13 |
| 200k | HPYLM | 24.22 | 52.295 | 22.32 | 105.22 |
| 200k | GTPYLM | 23.22 | 53.332 | 22.21 | 110.12 |
| size | system | FR-EN | Perplexity | EN-FR | Perplexity |
| 50k | baseline1 | 17.68 | 188.269 | 17.80 | 188.329 |
| 50k | baseline2 | 17.58 | 190.874 | 17.60 | 190.314 |
| 50k | HPYLM | 17.81 | 168.221 | 18.32 | 178.269 |
| 50k | GTPYLM | 17.01 | 178.303 | 18.33 | 179.200 |
| 200k | baseline1 | 18.40 | 162.573 | 18.20 | 165.839 |
| 200k | baseline2 | 18.19 | 165.232 | 18.02 | 168.989 |
| 200k | HPYLM | 18.99 | 148.338 | 18.60 | 153.921 |
| 200k | GTPYLM | 18.70 | 152.104 | 18.50 | 160.332 |
| size | system | ES-EN | Perplexity | EN-ES | Perplexity |
| 50k | baseline1 | 16.21 | 198.274 | 15.17 | 156.861 |
| 50k | baseline2 | 16.01 | 198.274 | 15.01 | 152.435 |
| 50k | HPYLM | 16.91 | 194.773 | 15.87 | 151.434 |
| 50k | GTPYLM | 16.68 | 196.403 | 15.75 | 153.224 |
| 200k | baseline1 | 16.87 | 168.431 | 17.62 | 154.273 |
| 200k | baseline2 | 16.37 | 174.856 | 17.32 | 168.754 |
| 200k | HPYLM | 17.50 | 152.312 | 18.20 | 145.223 |
| 200k | GTPYLM | 17.15 | 156.440 | 18.10 | 146.211 |

Table I
RESULTS FOR LANGUAGE MODEL. BASELINE1 IS BY MODIFIED KNESER-NEY METHOD AND BASELINE2 IS BY GOOD-TURING METHOD.

only to language models. However, it would be possible to apply a similar method to the translation model. Thirdly, we may extend our approach to syntax-based or dependency-based LMs.

## VI. ACKNOWLEDGMENTS

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (http://www.cngl.ie) at Dublin City University. We would also like to thank the Irish Centre for High-End Computing.

## REFERENCES

[1] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer-Verlag London. 2006.

[2] A. Fujii, M. Utiyama, M. Yamamoto, T. Utsuro, T. Ehara, H. Echizen-ya, S. Shimohata. "Overview of the Patent Translation Task at the NTCIR-8 Workshop". Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access. 2010. pp. 293–302.

[3] W. A. Gale. "Good-Turing Smoothing Without Tears", Journal of Quantitative Linguistics. 1995. 2:3, pp. 217–237.

[4] S. Goldwater, T. L. Griffiths, and M. Johnson. "Contextual dependencies in unsupervised word segmentation". In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING-ACL06). Sydney, Australia. July, 2006. pp. 673–680.

[5] S. Huang and S. Renals. "Hierarchical Bayesian Language Models for Conversational Speech Recognition". IEEE Transactions on Audio, Speech and Language Processing, 2009. 18:8, pp. 1941–1954.

[6] P. Koehn, "Statistical Significance Tests for Machine Translation Evaluation". In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain. 2004. pp. 388-395.

[7] P. Koehn. "Europarl: A Parallel Corpus for Statistical Machine Translation". In Proceedings of the 10th Machine Translation Summit (MT Summit X), Phuket, Thailand, Sep, 2005. pp. 79–86.

[8] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. "Moses: Open source toolkit for Statistical Machine Translation". In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, 2007. pp. 177–180.

[9] R. Kneser and H. Ney. 1995. "Improved backing-off for n-gram language modeling". In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, Detroit, MI, 1995, pp. 181–184.

[10] D. Mochihashi, T. Yamada and N. Ueda. "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling". In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009), Singapore, August, 2009. pp. 100–108.

[11] D. Mochihashi and E. Sumita. "The Infinite Markov Model". In Proceedings of the 20th Neural Information Processing Systems (NIPS 2007), Vancouver, 2007. pp. 1017–1024.

[12] F. Och and H. Ney. "A Systematic Comparison of Various Statistical Alignment Models". Computational Linguistics, 29:1, 2003. pp. 19-51.

[13] J. Pitman. "Exchangeable and partially Exchangeable Random Partitions". Probability Theory and Related Fields, Vol. 102, pp. 145-158, 1995.

[14] A. Stolcke. "SRILM – An extensible language modeling toolkit". In Proceedings of the International Conference on Spoken Language Processing. 2002. pp. 901–904.

[15] E. Sudderth and M. Jordan. "Shared Segmentation of Natural Scenes using Dependent Pitman-Yor Processes". In Proceedings of the 21th Neural Information Processing Systems (NIPS 2008), Vancouver, Dec, 2008. pp. 1585–1592.

[16] Y. W. Teh. "A hierarchical Bayesian language model based on Pitman-Yor processes". In Proceedings of Joint Conference of the 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 2006. pp. 985–992.

[17] V. Vapnik. "Statistical Learning Theory". John Wiley & Sons, 1998.