Pitman-Yor Process-Based Language Models for Machine Translation

Tsuyoshi Okita, Andy Way
Dublin City University, CNGL / School of Computing,
Glasnevin, Dublin 9, Ireland
{tokita, 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, language models, Kneser-Ney method, Chinese restaurant process.

1 Introduction

Statistical approaches or non-parametric Machine Learning methods estimate some targeted statistical quantities based on the (true) posterior distributions in a Bayesian manner (Bishop, 2006) or based on the underlying fixed but unknown (joint) distributions from which we assume that we sample our training examples in a frequentist manner (Vapnik, 1998). In NLP (Natural Language Processing), such distributions are observed by simply counting (joint / conditional) events, such as $c(w)$, $c(w_0, w_1, w_2)$ and $c(w_3 \mid w_1, w_2)$ where w denotes words and $c(\,)$ 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 (Gale, 1994; Kneser and Ney, 1995; Chen and Goodman, 1998). Most smoothing techniques do not have a statistical model but rely on some heuristics such as discounting, interpolation, and 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 (Goldwater et al., 2006, Teh, 2006). 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) (Teh, 2006; Mochihashi and Sumita, 2007; Huang and Renal, 2009) whose generative model uses hierarchies of n-grams. This model is shown to be superior to the interpolated Kneser-Ney methods (Kneser and Ney, 1995) and comparable to the modified Kneser-Ney methods (Chen and Goodman, 1998) in terms of perplexity. Hierarchical Pitman-Yor processes have been successfully applied to word segmentation as well (Goldwater et al., 06; Mochihashi et al., 2009).

This paper is organized as follows. Mentioning language model and perplexity in Section 2, Section 3 briefly reviews smoothing methods. Section 4 discusses HPYLM, Good-Turing Pitman-Yor language model (GTPYLM), and a minor change in the PB-SMT decoding algorithm. Experimental results are presented in Section 5. Section 6 concludes and provides avenues for further research.

2 Language Model and Perplexity

Let w_i denotes a word, and W denotes a sequence of words $w_1,w_2,\ldots,w_n$. A language model aims at modelling $p(W)$ ($= p(w_1,\ldots,w_m)$) such that $p(W)$ predicts the probability of picking up a sequence of words W. In an n-gram language model, the probability $p(w_1,\ldots,w_m)$ of observing the sentence $w_1,\ldots,w_m$ is approximated as in (1):

$$\begin{aligned} p(W) &= p(w_1,\ldots,w_m) \\ &= \prod_{i=1}^{m} p(w_i \mid w_1,\ldots,w_{i-1}) \\ &= \prod_{i=1}^{m} p(w_i \mid w_{i-m},\ldots,w_{i-1}) \end{aligned} \tag{1}$$

Note that $p(w_1,\ldots,w_n) = \prod p(w_1,\ldots,w_{i-1})$ holds by the product rule to express the joint distribution for a sequence of observations, and $P(w_n \mid w_1,\ldots,w_{n-1}) = P(w_n \mid w_{n-m},\ldots,w_{n-2},w_{n-1})$ holds by the Markov assumption of the history up to m words.

The measure to evaluate the performance of language model is often done by perplexity defined as in (2):

$$2^{\sum_{i=1}^{N} \frac{1}{N} \log_2 q(x_i)} \tag{2}$$

The perplexity suggests how well it predicts a separate test sample $x_1,\ldots,x_N$ also drawn from p, for a given a proposed model q.

3 Smoothing Methods

This section reviews various smoothing methods in the context of language model (Manning and Schutze, 1999; Jurafsky and Martin, 2009; Koehn, 2010). These are developed based on the heuristic combination of absolute discount, back-off, interpolation schemes, and so forth. Without loss of generality this subsection explains the difference of smoothing methods using the bi-gram language model. We use the notation: $w_{i-1}w_i$ denotes the consecutive two words, •w denotes the consecutive two words where the first word is any word, and *c( )* denotes a function which counts the words specified as its argument. The condition $c(w_{i-1}w_i) > 0$ means that the bi-gram $w_{i-1}w_i$ appeared in the corpus. (Hence, in most cases below, the condition *otherwise* means that the bi-gram $w_{i-1}w_i$ did not appear in the corpus.

We start with a maximum likelihood method, which is shown in (3).

$$p_{ML}(w_i \mid w_{i-1}) = \begin{cases} \dfrac{c(w_{i-1}w_i)}{\sum_w c(w_{i-1}w)} & if \quad c(w_{i-1}w) \\ 0 & otherwise \end{cases} \tag{3}$$

Since the maximum likelihood reflects purely statistics, there is no value assigned for unobserved n-grams, which is shown in *otherwise* in (3). If we subtracts a fixed (absolute) discount *D* from each count in order to allocate the mass for unobserved bi-grams, this is called the absolute discounting method, which is shown in (4):

$$p_{absolute}(w_i \mid w_{i-1}) = \begin{cases} \dfrac{c(w_{i-1}w_i) - D}{\sum_w c(w_{i-1}w)} & if \quad c(w_{i-1}w_i) > 0 \\ \alpha(w_i)\, p(w_i) & otherwise \end{cases} \tag{4}$$

If we take into account the diversity of histories for the unobserved bi-grams, this is called a Kneser-Ney smoothing method (Kneser and Ney, 1995). With the definition of the count of histories for a word as in (5),

$$N_{1+}(\bullet w) = \mid \{w_i : c(w_{i-1}w_i) > 0\} \mid \tag{5}$$

the raw counts of the maximum likelihood estimation is replaced with this count of histories for a word. In sum, a Kneser-Ney method is written as in (6):

$$p_{KN}(w_i \mid w_{i-1}) = \begin{cases} \dfrac{c(w_{i-1}w_i) - D}{\sum_w c(w_{i-1}w)} & if \quad c(w_{i-1}w_i) > 0 \\ \alpha(w_i) \dfrac{N_{1+}(\bullet w)}{\sum_w N_{1+}(w_i w)} & otherwise \end{cases} \tag{6}$$

If we combine the ideas behind interpolation and back-off, we can combine two terms in the right-hand side in (6). This is called an interpolated Kneser-Ney method (Chen and Goodman, 1998), which is shown in (7):

$$p_{\text{interpolatedKN}}(w_i \mid w_{i-1}) = \begin{cases} \frac{c(w_{i-1}w_i) - D}{\sum_w c(w_{i-1}w)} + \beta(w_i)\frac{N_{1+}(\bullet w)}{\sum_w N_{1+}(w_i w)} & c(w_{i-1}w_i) > 0 \\ \beta(w_i)\frac{N_{1+}(\bullet w)}{\sum_w N_{1+}(w_i w)} & otherwise \end{cases} \quad (7)$$

Now, if we have an intuition that an absolute discount D_n for each n-gram takes different values (but a fixed values) shown in (8),

$$D(n) = \begin{cases} D_1 \ (if\ c = 1) \\ D_2 \ (if\ c = 2) \\ D_{3+} \ (if\ c \geq 3) \end{cases} \quad (8)$$

this method makes a modified Kneser-Ney method (Chen and Goodman, 1998), which is shown in (9). Note that we derive (9) from (7) using (8) for different n-grams. Note that similarly with (7), although each distribution has the case when bi-gram is not observed it is omitted from (9).

$$\begin{cases} p_{\text{interpolatedKN\_unigram}}(w_i) = \frac{c(w_i) - D_1}{\sum_w c(w)} + \beta_1(w_i)\frac{N_{1+}(\bullet)}{\sum_w N_{1+}(w)} & (W = unigram) \\ p_{\text{interpolatedKN\_bigram}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}w_i) - D_2}{\sum_w c(w_{i-1}w)} + \beta_2(w_i)\frac{N_{1+}(\bullet w)}{\sum_w N_{1+}(w_i w)} & (W = bigram) \\ p_{\text{interpolatedKN\_trigram}}(w_i \mid w_{i-2}w_{i-1}) = \frac{c(w_{i-2}w_{i-1}w_i) - D_3}{\sum_w c(w_{i-2}w_{i-1}w)} + \beta_3(w_i)\frac{N_{1+}(\bullet w_{i-1}w)}{\sum_w N_{1+}(w_{i-2}w_i w)} & (W \geq trigram) \end{cases} \quad (9)$$

A Good-Turing method (Good, 1953) introduces the count-of-counts *Nc* shown in (10),

$$N_c = \sum_{x:count(x)=c} 1, \quad (10)$$

which is the number of different words that were seen exactly *c* times. Using this *Nc* this method infers the zero probability mass. Let *N* denotes the total number of counts. The modified count *c** can be obtained by

$$c^* = (c+1)\frac{N_{c+1}}{N_c} \quad (11)$$

Using these quantities, the probability mass for unobserved n-grams can be calculated as in (12) where the mass for unobserved n-grams are uniformly allocated:

$$p_{\text{GoodTuring}}(w_1, \ldots, w_n) = \begin{cases} \frac{c^*}{N} & if \quad c(w_1, \ldots, w_n) > 0 \\ \frac{1 - \sum_{i=1}^{\infty} c^* \frac{N_i}{N}}{N_0} & if \quad c(w_1, \ldots, w_n) = 0 \end{cases} \quad (12)$$

4 Hierarchical Pitman-Yor Language Model

This section describes the statistical smoothing method based on the hierarchical Pitman-Yor process, which is a non-parametric generalization of the Dirichlet distribution that produces power-law distributions (Teh 2006; Goldwater et al., 2006). Hence, this smoothing method of hierarchical Pitman-Yor processes does a smoothing task under the prior knowledge that the underlying distribution has power-law properties. Following descriptions are based on various literatures (Teh, 2006; Mochihashi and Sumita, 2007; Mochihashi et al., 2009; Huang and Renal, 2009).

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 n-grams, 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.

4.1 HPYLM: Generative Model

HPYLM is constructed in the following way encoding the property of the power-law distribution. A graphical model of HPYLM is shown in Figure 1.

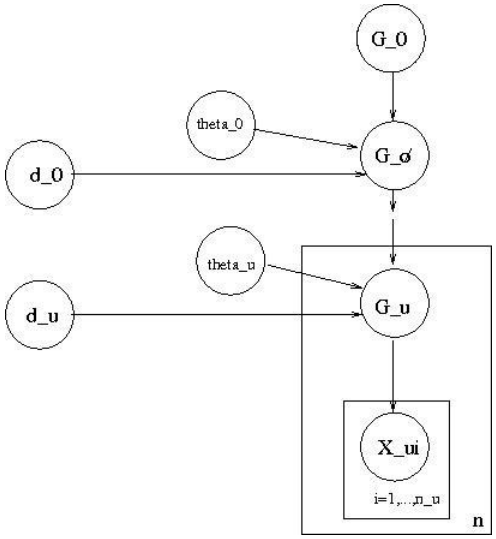

Figure 1: Graphical model of hierarchical Pitman-Yor language model.

Firstly, we use a Pitman-Yor process. The Pitman-Yor process $PY(d, \theta, G_0)$ is a distribution over a base distribution G_0 having two parameters d and θ where d is called a

discount parameter (which is also called a concentration parameter) and θ is called a strength parameter (Pitman, 1995). A strength parameter $\theta > -d$ controls the degree to which the new draw is assigned to among those which are not appeared in the past, while a discount parameter $0 \leq d < 1$ specifies the degrees to which the new draw resembles the base distribution G_0. As far as $0 < d < 1$ the Pitman-Yor process $PY(d, \theta, G_0)$ poses a characteristic to generate a power-law distribution, while $d = 0$ the Pitman-Yor process reduces to a Dirichlet process. The characteristic of a power-law distribution is among one of the motivation of this generative procedure. A power-law phenomenon can be characterized by those two: the more words have been assigned to a draw from G_0, the more likely subsequent words will be assigned to the draw (the richer-gets-richer property), while the more we draw from G_0, the more likely a new word will be assigned to a new draw from G_0.

Secondly, we place a Pitman-Yor process as a prior in this generative model. Let u be a given context, $\pi(u)$ be a function whose parameter is a context u, $d_{|u|}$ be a discount parameter of the length $|u|$ of its context, $\theta_{|u|}$ be a strength parameter of the length $|u|$ of its context, and $G_u(w)$ be the probability of the current word taking value w for a given context u. Using a Pitman-Yor process as the prior for as in (13):

$$G_u \mid d_{|u|}, \theta_{|u|}, G_{\pi(u)} \sim PY(d_{|u|}, \theta_{|u|}, G_{\pi(u)}) \tag{13}$$

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.

Thirdly, $\pi(u)$ is defined as the suffix of u consisting of all but the earliest word in Equation (1) as (Teh, 2006). This signifies that u is n-gram words and $\pi(u)$ is (n-1)-gram words; this induction of Equation (1) makes an n-gram hierarchy.

Fourthly, as the last sentence suggests, such a prior of the Pitman-Yor processes can be placed recursively over $G_{\pi(u)}$ in the generative model, with a base distribution G_0 sharing across the different Pitman-Yor processes G_j, as in Equation (14):

$$\begin{cases} G_u \mid d_{|u|}, \theta_{|u|}, G_{\pi(u)} \sim PY(d_{|u|}, \theta_{|u|}, G_{\pi(u)}) \\ G_\emptyset \mid d_0, \theta_0, G_0 \sim PY(d_0, \theta_0, G_0) \end{cases} \tag{14}$$

This is repeated until we get to G_0, the vector of probabilities over the current word given the empty context $\emptyset$. Let W be a fixed and finite vocabulary of V words. G_0 is the global mean vector, given a uniform value of $G_0 = 1 / V$ for all $w \in W$.

4.2 HPYLM: Inference

One procedure to do an inference in order to generate words drawn from G is called Chinese restaurant process, which iteratively marginalizes out G. Note that when the vocabulary is finite, $PY(d, \theta, G_0)$ has no known analytic form.

We assume the language modelling. Let h be an n-gram context; for example in 3-gram, this is $h = \{w_1, w_2\}$. A Chinese restaurant contains an infinite number of table t, each with infinite seating capacity. Customers, which are the n-gram counts $c(w|h)$, enter the restaurant and seat themselves over the tables $1, ..., t_{hw\bullet}$. 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 $c_{hwk} - d$, or at a new unoccupied table with probability proportional to $\theta + dtk_\bullet$ as is shown in (3):

$$w \mid h \sim \begin{cases} c_{hwk} - d \,(1 \le k \le t_{hw}) \\ \theta + dt_h \quad (k = new) \end{cases} \tag{15}$$

where c_{hwk} is the number of customers seated at table k until now, and $t_{k\bullet} = \Sigma_w t_{kw}$ is the total number of tables in h. An occupied table of the first line in (15) corresponds to a Dirichlet process where the parameter d specifies the degree to which the distribution resembles the base distribution, and an unoccupied table of the second line in (15) corresponds to a Poisson process where the parameter θ controls the rate of allocation of a new draw.

Hence, the predictive distribution of n-gram probability in HPYLM is recursively calculated as in Equation (16):

$$p(w \mid h) = \frac{c(w \mid h) - dt_{kw}}{\theta + c(h)} + \frac{\theta + dt_{k\bullet}}{\theta + c(h)} p(w \mid h') \tag{16}$$

where $p(w|h')$ is the same probability using a (n-1)-gram context h'. The case when $t_{kw} = 1$ corresponds to an interpolated Kneser-Ney smoothing (Chen and Goodman, 1998).

Implementation of this inference procedure relates to the Markov chain Monte Carlo (MCMC) sampling methods. Hence, for a given n-gram hierarchies from a training set, d and θ (as well as G_j) are solved as a non-parametric Machine Learning method. The simplest way 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 (Goldwater et al., 2006). A blocked Gibbs sampler is proposed by Mochihashi et al. (Mochihashi et al., 2009), 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.

Note that the case when $t_{kw} = 1$ corresponds to an interpolated Kneser-Ney smoothing (Chen and Goodman, 1998). Teh explains this in this way (Teh, 2006): If we restrict $t_{hw\bullet}$ to be at most 1 as in (17):

$$t_{hw\bullet} = \min(1, c_{hw\bullet}) \tag{17}$$

we will obtain the same discount value so long as $c_{hw\bullet} > 0$, i.e. absolute discounting. Furthermore, supposing that the strength parameters are all $\theta_{|k|} = 0$, the predictive probabilities in Equation (5) now directly reduce to the predictive probabilities given by interpolated Kneser-Ney smoothing method (Chen and Goodman, 1998).

4.3 Good-Turing Pitman-Yor Language Model

We use the same generative model which uses the Pitman-Yor process as a prior in Equation (2) once (not recursively), and let us now consider $\pi(u)$ as a count-counts function. (This is also known as event-counts or count of counts.) We refer to this model as Good-Turing Pitman-Yor Language Model (GTPYLM). Our intention here is to incorporate a prior knowledge that a distribution takes a power-law distribution, as well as incorporating the zero-frequency mass.

We use the notation of (10) and (11). Then, by (14), the predictive distribution of n-gram probability in GTPYLM is computed as in (18):

$$p^*(w_i \mid w_{i-1}, N_w) = \frac{c^*(w \mid w_{i-1}, N_w) - dt_{N_w w}}{\theta + c^*(w \mid N_w)} + \frac{\theta + dt_{k\bullet}}{\theta + c^*(w \mid N_w)} p^*(w_i \mid w_{i-1} N_{w-1}) \quad (18)$$

Note that this formulation does not avoid the problem of data sparseness of N_c when c is large which requires to obtain in the similar way as other literatures, such as Gale (1994).

4.4 PB-SMT Decoding Algorithm for HPYLM and GTPYLM

A minor difference in the decoding process is required. In a test sentence, if we encounter unseen n-grams, 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, θ, and so forth. Such parameters enable us to calculate the zero-probabilities for unseen n-grams in a test sentence. The overall algorithm to obtain updated HPYLM is shown in Algorithm 1.

1. Given: a test sentence, HPYLM *p(w)*, HPYLM parameter file *p0(w)*.
2. By generating possible n-gram candidates, using *p0(w)* we update HPYLM *p'(w)*.
3. Run a decoder which looks up the updated HPYLM *p'(w)*.

Algorithm 1. Decoding algorithm for HPYLM p(w)

5 Experimental Results

5.1 Experimental Setup

For all the experiments, we used a standard log-linear phrase-based MT system based on Moses (Koehn et al., 2007). The GIZA++ implementation (Och and Ney, 2003) of IBM Model 4 was used for word alignment, followed by the grow-diag-final heuristics as phrase extraction. We used SRILM (Stolcke, 2002) to derive a 5-gram language model. We performed MERT (Och, 2003) and use a Moses decoder (Koehn et al., 2007). The baseline 1 derived a 5-gram language model by SRILM with modified Kneser-Ney method (Chen and Goodman, 1998) and the baseline 2 used with SRILM with Good-Turing method (Good, 1953).

For the HPYLM and GTPYLM, we obtained the results by a method using a blocked Gibbs sampler (Mochihashi et al., 2009), which was considerably more efficient compared to a conventional Gibbs sampler (Goldwater et al., 2006; Teh, 2006). In this experiment, we used a phrase table derived by the conventional method. Perplexity was measured in terms

of the same test set.

5.2 Experimental Results

We conducted an experimental evaluation for JP-EN on the NTCIR-8 corpus (Fujii et al., 2010; Okita et al., 2010d) and for FR-EN and ES-EN on Europarl (Koehn, 2005). We randomly extracted two training corpora of 50k and 200k sentence pairs, where we used

size	system	EN-JP	perplexity	JP-EN	perplexity
50k	baseline1	16.33	71.460	22.01	131.438
50k	baseline2	16.20	72.435	22.81	136.812
50k	HPYLM	17.36	66.012	22.81	116.074
50k	GTPYLM	17.27	67.112	22.70	120.320
200k	baseline1	23.42	59.607	21.68	117.780
200k	baseline2	23.36	58.587	21.38	119.130
200k	HPYLM	24.22	52.295	22.32	105.220
200k	GTPYLM	23.22	53.332	22.21	110.120

Table 1. Performance between EN-JP.

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

Table 2. Performance between FR-EN.

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 3. Performance between ES-EN.

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 1, 2, and 3. 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 re-sampling (Koehn, 04). 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.

6 Conclusion

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, although our experiments are limited only to language models, it would be possible to apply the similar smoothing method to the translation model which reflects prior knowledge. In

our context, we add prior knowledge into the translation model by a Multi-Word Expression-sensitive word aligner (Okita et al., 2010a) and by a noise reduced word aligner (Okita, 2009; Okita et al., 2010b). We believe that in such cases, smoothing methods are necessary because the surface of posterior probability is considered to be perturbed by such prior knowledge. The preliminary results have been obtained for the translation model enhanced with prior knowledge such as Multi-Word Expressions, paraphrases and Out-Of-Vocabulary words (Okita and Way, 2010c). We used the smoothing method similar to HPYLM for translation model and obtained slightly better results for small corpus between EN and JP. Thirdly, we may extend our approach to syntax-based or dependency-based language models.

7 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.

8 References

Akaike, H.. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control,* **19** (6), pp. 716–723.

Bishop, C. M.. 2006. Pattern Recognition and Machine Learning. *Springer-Verlag* London.

Boser, B. E., Guyon, I. M., and Vapnik, V. N.. 1992. A training algorithm for optimal margin classifier. *D. Haussler, editor, In 5th Annual ACM Workshop on COLT,* Pittsburgh, PA, ACM Press. pp. 144-152.

Chen, S. and Goodman, J.. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report TR-10-98, Harvard University,* August. pp. 1-63.

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

Gale, W.A., 1995. Good-Turing Smoothing Without Tears. *Journal of Quantitative Linguistics*. vol. 2, no. 3, pp. 217-237.

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

Good, I. J., 1953. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika,* vol. 40, no. 16, pp. 237-264.

Jurafsky, D., and Martin, J. H., 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. *Pearson International Edition*.

Huang, S., and Renals, S, 2009. Hierarchical Bayesian Language Models for Conversational Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 8, pp. 1941-1954.

Koehn, P., 2004. Statistical Significance Tests for Machine Translation Evaluation. *In*

Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain. pp. 388-395.

Koehn, P., 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *In Proceedings of the 10th Machine Translation Summit (MT Summit X)*, Phuket, Thailand, pp. 79-86.

Koehn, P., 2010. Statistical Machine Translation. *Cambridge University Press*, London.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E., 2007. 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, pp. 177-180.

Kneser, R., and Ney, H., 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, pp. 181-184.

Mochihashi, D., Yamada, T., and Ueda, N., 2009. 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, pp. 100-108.

Manning, K. D., and Schutze, H., 1999. Foundations of Statistical Natural Language Processing. *The MIT Press*, Cambridge, Massachusetts.

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

Neal, R. M., 1991. Bayesian mixture modeling by Monte Carlo simulation, *Technical Report CRG-TR-91-2, Dept. of Computer Science, University of Toronto*. pp. 1-23.

Och, F., 2003. Minimum Error Rate Training in Statistical Machine Translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics,* Sapporo, Japan. pp. 160-167.

Och, F., and Ney, H., 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, vol. 29, no. 1, pp. 19-51.

Okita, T., 2009. Data Cleaning for Word Alignment. *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), Student Research Workshop*, Singapore. pp 72-80.

Okita, T., Maldonado Guerra, A., Graham, Y., and Way, A., 2010a. Multi-Word Expression-Sensitive Word Alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010, collocated with COLING2010)*, Beijing, China. pp. 26-34.

Okita, T., Graham, Y., and Way, A., 2010b. Gap Between Theory and Practice: Noise Sensitive Word Alignment in Machine Translation. *In Journal of Machine Learning Research Workshop and Conference Proceedings Volume 11: Workshop on Applications of Pattern Analysis (WAPA2010).* Cumberland Lodge, England. pp. 119-126.

Okita, T., and Way, A., 2010c. Statistical Machine Translation with Terminology. *In*

Proceedings of the First Symposium on Patent Information Processing (SPIP), Tokyo, Japan. pp. 1-8.

Okita, T., Jiang, J., Haque, R., Al-Maghout, H., Du, J., Naskar, S. K., and Way, A., 2010d. MaTrEx: the DCU MT System for NTCIR-8. *In Proceedings of the NII Test Collection for IR Systems-8 Meeting (NTCIR-8),* Tokyo, Japan. pp. 377-383.

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

Rissanen, J., 1978. Modeling by shortest data description. *Automatica*, vol. 14, pp. 465-471.

Schwarz, G. E., 1978. Estimating the dimension of a model. *Annals of Statistics, vol.* 6, no. 2, pp. 461–464.

Stolcke, A., 2002. SRILM - An extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing*. pp. 901-904.

Sudderth, E., and Jordan, M., 2008. Shared Segmentation of Natural Scenes using Dependent Pitman-Yor Processes. *In Proceedings of the 21th Neural Information Processing Systems (NIPS2008),* Vancouver. pp. 1585-1592.

Teh, Y. W., 2006. 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. pp. 985-992.

Vapnik, V., 1998. Statistical Learning Theory. John Wiley & Sons, London.

Appendix.A. Discussion about Model Complexity: Model Selection Method and Smoothing Method

Modern Machine Learning algorithms, such as Support Vector Machines (Boser et al., 1992; Vapnik, 1998), seek to obtain small generalization error over unseen data. This mechanism is implemented by minimizing both of the risk and the capacity of the function class automatically. Suppose that we are not able to adjust generalization error automatically. Instead, we use some measure to evaluate model complexity, such as An Information Criterion (or Akaike Information Criterion; AIC) (Akaike, 1974), Bayesian Information Criterion (BIC) (Schwarz, 1978), and Minimal Description Length (MDL) (Rissanen, 1978). Now, we start with some model complexity and try to adjust this model complexity for a given data in order to obtain the best generalization error over unseen data. If the initial model complexity is under the point which achieves the best generalization error (or an equilibrium point), this is called `under-fitting' (a point A3 in Figure 2). If the initial model complexity is beyond the point which achieves the best generalization error, this is called `over-fitting' (a point A4 in Figure 2). Hence, we can say that a model selection technique aims at transferring A3 or A4 into an equilibrium state at A2.

Similarly, we may explain the statistical smoothing technique of hierarchical Pitman-Yor process using this figure. Firstly, the mass for zero probabilities which are the results of this smoothing technique are all virtual samples which are not existed in training data. Hence the arrow is from left to right. Secondly, under unsupervised learning, we may use the knowledge that an infinite mixture models will lead to the best number of clusters (Neal, 1991), or at least trying to achieve such best configuration, at a point with a small generalization error (or with small energy). Since the parameters which are targeted in a (hierarchical) Pitman-Yor process method are two, a strength parameter θ and a discount

parameter d, for each process, the model complexity will not change so long as the n-gram hierarchies are not augmented. Hence, this situation can be depicted as the right arrow from the point A1 to A2. It is noted that in language modelling, if we see the number of words m as the number of parameters, the complexity of this n-gram system becomes O(mn). However, the statistical smoothing methods do not treat these n-gram words as parameters.

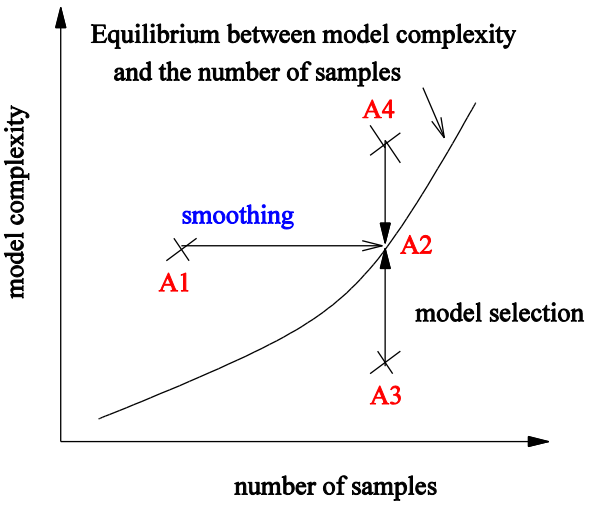

Figure 2: Schematic explanation of difference between smoothing methods and model selection methods.