This paper proposes a hierarchical Pitman-Yor process language model. This direct generalization of hierarchical Dirichlet language model using Pitman-Yor process as priors has an efficient inference scheme, gives performance superior to stat-of-the-art Kneser-Ney smoothing methods, and also builds up the direct correspondence between Bayesian approach and interpolated Kneser-Ney.

The popular n-gram models for language models usually have tremendous parameters to train. In contrast, the hierarchical Pitman-Yor process has only 2n parameters for a n-gram model. It is based on a hierarchical Bayesian model where each hidden variable is distributed per Pitman-Yor process which is a nonparametric (Wikipedia, “either being distribution-free or having a specified distribution but with the distribution's parameters unspecified”) generalization of Dirichlet distribution. For a Pitman-Yor process prior G, there are 3 parameters: discount parameter d, strength parameter and mean vector G0. Notice that this process has 2 self-reinforcing effects which produce a power-law distribution with many unique words. As a metaphor, Chinese restaurant process considers a sequence of customers (the words draw from G) and an unbounded number of tables (the draws from G0) capable to accommodate infinite customers: the first customer sits at the first table, and then each subsequent customer either joins an already occupied table (assign value of a previous draw) or occupies a new table (assign value of a new draw).

The proposed model is an n-gram language model based on a hierarchical extension of the Pitman-Yor process. It is important to find the conditional probabilities of the current word given its various contexts. For the probability vector G**u** of the current word given a context **u**, it is placed a Pitman-Yor process prior taking as mean vector the probability vector G of the same current word given all but the earliest word in the context (suffix). This recursive generative process is repeated until there is empty context where the uniform prior is applied. With all probability vector G**u**’s marginalized out, there is another corresponding analogous Chinese restaurant process (seating arrangement S**u**: assignment of words x**u**l to draws y**u**k from “parent” distribution), which is amendable to efficient inference and easy computation of predictive probabilities. Similar to the “rich-gets-richer” clustering property of the original Pitman-Yor process, the hierarchical Pitman-Yor process also has a self-reinforcing property: the more a word w has been drawn in the context **u**, the more likely will word w be drawn again in **u**.

Then it comes to inference schemes (based on Markov chain Monte Carlo sampling). With equivalent posterior over seating arrangements, the predictive probabilities of a test word w after a context **u** is of interest. Gibbs sampling is used to obtain the posterior samples of hyperparameter {contexts, parameters} pairs. And the conditional probabilities of a word w given context **u** under certain hyperparameters is estimated recursively according to hierarchical Chinese restaurant process. As for costs, this sampling scheme takes time proportional to the number of words in the training set and space proportional to the number of unique n-grams. In addition, the growth of discounts is sublinear on average. Furthermore, with restriction on parameters t and , the formula of predictive probabilities directly reduces to that from interpolated Kneser-Ney method, which relates Bayesian approach and interpolated Kneser-Ney.

Experiments are conducted to compare hierarchical Pitman-Yor language model trained using a Gibbs sampler(HPYLM, and a modified version using parameters obtained from IKN with loose restriction using cross-validation, HPYCV), interpolated Kneser-Ney(IKN), modified Kneser-Ney(MKN) and the hierarchical Dirichlet language model (HDLM). Using perplexity as the evaluation metrics, the hierarchical Pitman-Yor language model outperforms other methods. Moreover, for cross-entropies in terms of how many times each word appears in the test set, HPYLM and HPYCV display different behaviors when compared with MKN.