A Systematic Bayesian Treatment of the IBM Alignment Models

Yarin Gal and Phil Blunsom

yg279@cam.ac.uk

June 12, 2013

Introduction

The IBM alignment models have underpinned the majority of statistical machine translation systems for almost twenty years.

- They offer principled probabilistic formulation and (mostly) tractable inference
- There are many open source packages implementing them
  - Giza++ – one of the dominant implementations,
  - employs a variety of exact and approximate EM algorithms

However –

Introduction

However –

- They use a parametric approach
    - Significant number of parameters to be tuned
- Intractable summations over alignments for models 3 and 4
    - Usually approximated using restricted alignment neighbourhoods
    - Shown to return alignments with probabilities well below the true maxima
- Sparse contexts are not handled
    - The models use weak smoothing interpolating with the uniform distribution

Many alternative approaches to word alignment have been proposed, and largely failed to dislodge the IBM approach.

Introduction

How can we overcome these problems?

- Use a different inference technique
  - Gibbs sampling
- Use non-parametric priors over the generative models
  - Replace the categorical distributions with others; for example, hierarchical Pitman-Yor processes

The Pitman-Yor process

We can define the *Pitman-Yor process* by describing how to draw from it:

The Pitman-Yor process: definition

Draws from the Pitman-Yor process $G_1 \sim PY(d, \theta, G_0)$ with a discount parameter $0 \leq d < 1$, a strength parameter $\theta > -d$, and a base distribution G_0, are constructed using a Chinese restaurant process as follows:

$$X_{c_{\cdot}+1}|X_1, ..., X_{c_{\cdot}} \sim \sum_{k=1}^{t_{\cdot}} \frac{c_k - d}{\theta + c_{\cdot}} \delta_{y_k} + \frac{\theta + t_{\cdot} d}{\theta + c_{\cdot}} G_0$$

Where c_k denotes the number of X_is (tokens) assigned to y_k (a type) and $t_{\cdot}$ is the total number of y_ks drawn from G_0.

- Successful in many latent variable language tasks

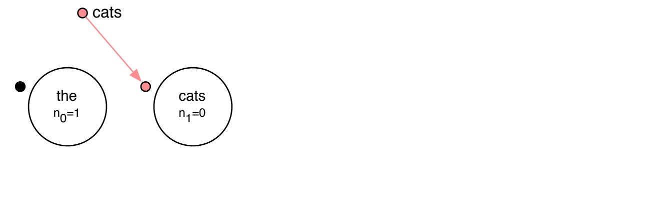
The Pitman-Yor process

The Chinese restaurant process

$$X_1 \sim G_0$$

The Pitman-Yor process

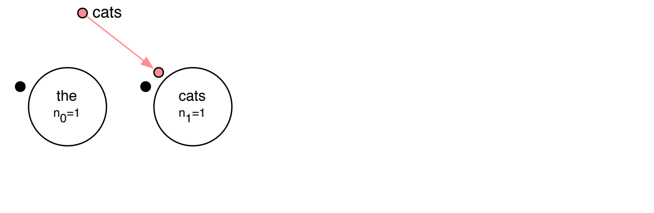
The Chinese restaurant process



$$X_2|X_1 \sim \frac{1-d}{\theta+1}\delta_{y_{the}} + \frac{\theta+d}{\theta+1}G_0$$

The Pitman-Yor process

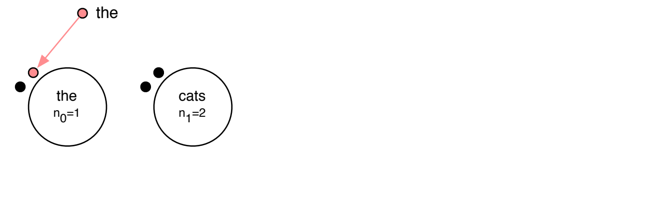
The Chinese restaurant process



$$X_3|X_1, X_2 \sim \frac{1-d}{\theta+2}\delta_{y_{the}} + \frac{1-d}{\theta+2}\delta_{y_{cats}} + \frac{\theta+2d}{\theta+2}G_0$$

The Pitman-Yor process

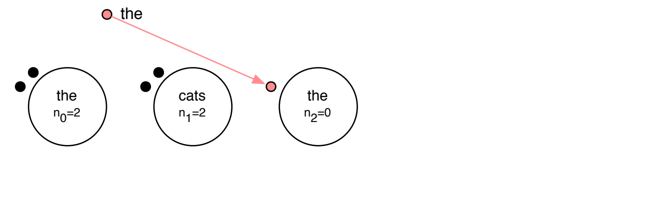
The Chinese restaurant process



$$X_4|X_1, X_2, X_3 \sim \frac{1-d}{\theta+3}\delta_{y_{the}} + \frac{2-d}{\theta+3}\delta_{y_{cats}} + \frac{\theta+2d}{\theta+3}G_0$$

The Pitman-Yor process

The Chinese restaurant process



$$X_5|X_1,...,X_4 \sim \frac{2-d}{\theta+4}\delta_{y_{the}} + \frac{2-d}{\theta+4}\delta_{y_{cats}} + \frac{\theta+2d}{\theta+4}G_0$$

The hierarchical Pitman-Yor process

The *hierarchical* Pitman-Yor process is simply a Pitman-Yor process where the base distribution is itself a Pitman-Yor process.

The hierarchical Pitman-Yor process: definition

$$
\begin{aligned}
w_i &\sim G_{\mathbf{u}} \\
G_{\mathbf{u}} &\sim PY(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \\
G_{\pi(\mathbf{u})} &\sim PY(d_{|\mathbf{u}|-1}, \theta_{|\mathbf{u}|-1}, G_{\pi(\pi(\mathbf{u}))}) \\
&\dots \\
G_{(w_{i-1})} &\sim PY(d_1, \theta_1, G_{\emptyset}) \\
G_{\emptyset} &\sim PY(d_0, \theta_0, G_0)
\end{aligned}
$$

where $|\mathbf{u}|$ denotes the length of context $\mathbf{u}$, $\pi(\mathbf{u})$ is obtained by removing the left most word, and G_0 is a base distribution (usually uniform over all words).

The hierarchical Pitman-Yor process

Comparing this to interpolated Kneser-Ney discounting language model, we see that Kneser-Ney is simply a hierarchical Pitman-Yor process with parameter θ set to zero and a constraint of one table $t_{\mathbf{u}w} = 1$:

Interpolated Kneser-Ney discounting language model

$$P_{\mathbf{u}}(w) = \frac{\max(0, c_{\mathbf{u}w} - d_{|\mathbf{u}|})}{c_{\mathbf{u}\cdot}} + \frac{d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}}{c_{\mathbf{u}\cdot}} P_{\pi(\mathbf{u})}(w)$$

The hierarchical Pitman-Yor process

$$P_{\mathbf{u}}(w) = \frac{c_{\mathbf{u}w} - d_{|\mathbf{u}|} t_{\mathbf{u}w}}{\theta + c_{\mathbf{u}\cdot}} + \frac{\theta + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}}{\theta + c_{\mathbf{u}\cdot}} P_{\pi(\mathbf{u})}(w)$$

- Shorter contexts are interpolated and have higher weight in the interpolation if the long context is sparse
- This view gives us a principled way of dealing with latent variables

IBM models - reminder

We can take advantage of the smoothing and interpolation with shorter contexts properties of the hierarchical Pitman-Yor (PY) process, and use it in word alignment.

Reminder: Model 1 generative story

$$P(F, A|E) = p(m|l) \times \prod_{i=1}^{m} p(a_i) p(f_i|e_{a_i})$$

Where $p(a_i) = \frac{1}{l+1}$ is uniform over all alignments and $p(f_i|e_{a_i}) \sim \mathit{Categorical}$.

- F and E are the input (source) and output (target) sentences of lengths J and I respectively,
- A is a vector of length J consisting of integer indices into the target sentence – the alignment.

PY-IBM model

Following the original generative story, we can re-formulate the model to use the hierarchical PY process instead of the categorical distributions:

PY Model 1 generative story

$$
\begin{aligned}
a_i | m &\sim G_0^m \\
f_i | e_{a_i} &\sim H_{e_{a_i}} \\
H_{e_{a_i}} &\sim PY(H_\emptyset) \\
H_\emptyset &\sim PY(H_0)
\end{aligned}
$$

- f_i and a_i are the i'th foreign word and its alignment position,
- e_{a_i} is the English word corresponding to alignment position a_i,
- m is the lengths of the foreign sentence.

IBM models - reminder

Extending this approach, we can re-formulate the HMM alignment model as well to use the hierarchical PY process instead of the categorical distributions.

Reminder: HMM alignment model generative story

$$P(F,A|E) =$$

$$p(m|l) \times \prod_{i=1}^{m} p(a_i|a_{i-1}, m) \times p(f_i|e_{a_i})$$

- f_i and a_i are the i'th foreign word and its alignment position,
- e_{a_i} is the English word corresponding to alignment position a_i,
- m and l are the lengths of the foreign and English sentences respectively.

PY-IBM model

We replace the categorical distribution for the transition $p(a_i|a_{i-1}, m)$ with a hierarchical PY process

PY HMM alignment model generative story

$$\begin{aligned} a_i|a_{i-1}, m &\sim G^m_{a_{i-1}} \\ G^m_{a_{i-1}} &\sim PY(G^m_{\emptyset}) \\ G^m_{\emptyset} &\sim PY(G^m_0) \end{aligned}$$

- Unique distribution for each foreign sentence length
- Condition the position on the previous alignment position, backing-off to the HMM's stationary distribution over alignment positions

IBM models - reminder

Reminder: Models 3 and 4 generative story

- We treat the alignment as a function from the source sentence positions i to $B_i \subset \{1, ..., m\}$ where the B_i's form a partition of the set $\{1, ..., m\}$,
- We define the fertility of the English word i to be $\phi_i = |B_i|$, the number of foreign words it generated,
- And $B_{i,k}$ refers to the kth word of B_i from left to right.

$$P(F, A|E) = p(B_0|B_1, ..., B_l) \times \prod_{i=1}^{l} p(B_i|B_{i-1}, e_i) \times \prod_{i=0}^{l} \prod_{j \in B_i} p(f_j|e_i)$$

IBM models - reminder

Reminder: Models 3 and 4 generative story

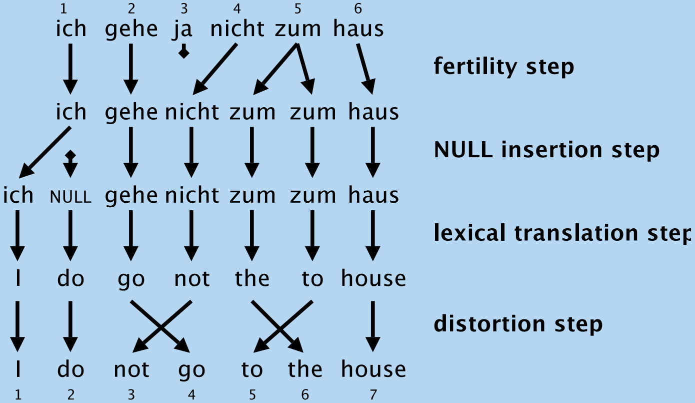
For model 3 the dependence on previous alignment sets is ignored and the probability $p(B_i|B_{i-1}, e_i)$ is modelled as

$$p(B_i|B_{i-1}, e_i) = p(\phi_i|e_i)\phi_i! \prod_{j \in B_i} p(j|i, m),$$

whereas in model 4 it is modelled using two HMMs:

$$\begin{aligned} p(B_i|B_{i-1}, e_i) = & p(\phi_i|e_i) \times p_{=1}(B_{i,1} - \odot(B_{i-1})|\cdot) \\ & \times \prod_{k=2}^{\phi_i} p_{>1}(B_{i,k} - B_{i,k-1}|\cdot) \end{aligned}$$

Models 3 and 4 word alignment [1]



[1] Borrowed from Philipp Koehn http://homepages.inf.ed.ac.uk/pkoehn/

PY-IBM model

Unlike previous approaches that ran into difficulties extending models 3 and 4, we can extend them rather easily by just replacing the categorical distributions.

- The inference method that we use, Gibbs sampling, circumvents the intractable sum approximation of other inference methods
- The use of the hierarchical PY process allows us to incorporate phrasal dependencies into the distribution
- We follow the original generative stories and extend them

PY-IBM model

Replacing the categorical priors with hierarchical PY process ones, we set the translation and fertility probabilities $p(\phi_i|e_i)\prod_{j\in B_i} p(f_j|e_i)$ using a common prior that generates translation sequences.

PY models 3 and 4 generative story

$$(f^1, ..., f^{\phi_i})|e_i \sim H_{e_i}$$

$$H_{e_i} \sim PY(H^{FT}_{e_i})$$

$$H^{FT}_{e_i}((f^1, ..., f^{\phi_i})) = H^{F}_{e_i}(\phi_i)\prod_j H^{T}_{(f^{j-1}, e_i)}(f^j)$$

- We used superscripts for the indexing of words which do not have to occur sequentially in the sentence

PY-IBM model

Replacing the categorical priors with hierarchical PY process ones, we set the translation and fertility probabilities $p(\phi_i|e_i)\prod_{j\in B_i} p(f_j|e_i)$ using a common prior that generates translation sequences.

PY models 3 and 4 generative story

$$\boxed{(f^{\,1}, ..., f^{\,\phi_i})|e_i \sim H_{e_i}}$$

$$H_{e_i} \sim PY(H^{FT}_{e_i})$$

$$H^{FT}_{e_i}((f^{\,1}, ..., f^{\,\phi_i})) = H^{F}_{e_i}(\phi_i)\prod_j H^{T}_{(f^{\,j-1},e_i)}(f^{\,j})$$

- We used superscripts for the indexing of words which do not have to occur sequentially in the sentence

PY-IBM model

Replacing the categorical priors with hierarchical PY process ones, we set the translation and fertility probabilities $p(\phi_i|e_i)\prod_{j\in B_i} p(f_j|e_i)$ using a common prior that generates translation sequences.

PY models 3 and 4 generative story

$$(f^1, ..., f^{\phi_i})|e_i \sim H_{e_i}$$

$$\boxed{H_{e_i} \sim PY(H^{FT}_{e_i})}$$

$$H^{FT}_{e_i}((f^1, ..., f^{\phi_i})) = H^F_{e_i}(\phi_i)\prod_j H^T_{(f^{j-1}, e_i)}(f^j)$$

- We used superscripts for the indexing of words which do not have to occur sequentially in the sentence

PY-IBM model

Replacing the categorical priors with hierarchical PY process ones, we set the translation and fertility probabilities $p(\phi_i|e_i)\prod_{j\in B_i} p(f_j|e_i)$ using a common prior that generates translation sequences.

PY models 3 and 4 generative story

$$(f^1, ..., f^{\phi_i})|e_i \sim H_{e_i}$$

$$H_{e_i} \sim PY(H_{e_i}^{FT})$$

$$\boxed{H_{e_i}^{FT}((f^1, ..., f^{\phi_i})) = H_{e_i}^{F}(\phi_i)\prod_j H^{T}_{(f^{j-1},e_i)}(f^j)}$$

- We used superscripts for the indexing of words which do not have to occur sequentially in the sentence

PY-IBM model

We generate sequences instead of individual words and fertilities, and fall-back onto these only in sparse cases.

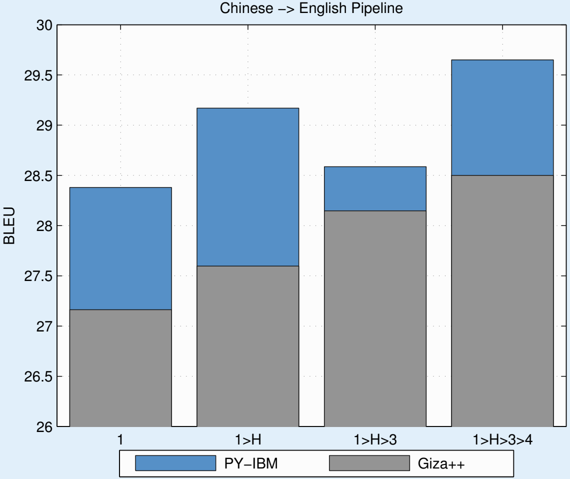
Example

Aligning the English sentence “I don’t speak French” to its French translation “Je ne parle pas français”, the word “not” will generate the phrase (“ne”, “pas”), which will later on be distorted into its place around the verb.

- The distortion probability for model 3, $p(j|i, m)$, is modelled as depending on the position of the source word i and its class
  - Interpolating for sparsity
  - The same way the HMM model backs-off to shorter sequences
- Similarly for the two HMMs in model 4.

Experiments

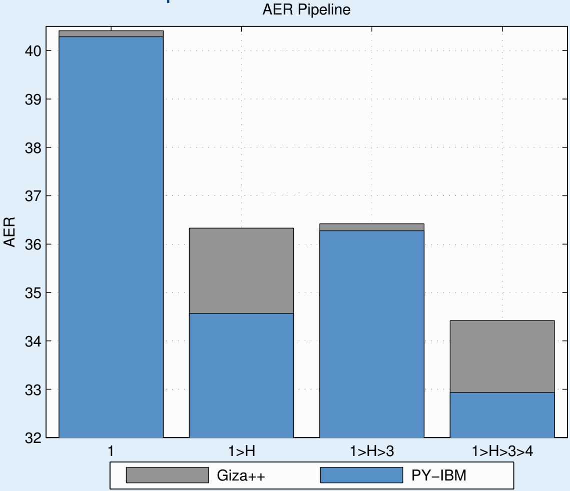
How does this model compare to the EM trained models?



BLEU scores of pipelined Giza++ and pipelined PY-IBM translating from Chinese into English on the FBIS corpus

Experiments

How does this model compare to the EM trained models?



AER of pipelined Giza++ and pipelined PY-IBM aligning Chinese and English on the FBIS corpus

Conclusions

- The models achieved a significant improvement in BLEU scores and AER on the tested corpus
- Follows the original generative stories while introducing additional phrasal conditioning into models 3 and 4
- Easy to extend and to introduce new dependencies without running into sparsity problems
  - Extension of the transition history used in the HMM alignment model
  - Introduction of dependencies on the context words and their part-of-speech information
  - Introduction of longer dependencies in the fertility and distortion distributions

Conclusions

We still need to do –

- Find more effective inference algorithms for hierarchical PY process based models
  - On bi-corpora limited in size (∼500K sentence pairs) the training currently takes 12 hours, compared to one hour for the EM models
  - More suitable for language pairs with high divergence – captures information that is otherwise lost
  - Recent research (e.g. Williamson, Dubey, and Xing 2013) provides good solutions for distributing collapsed samplers.

The PY-IBM models were implemented within the Giza++ code base, and are available as an open source package for further development and research at

`github.com/yaringal/Giza-Sharp`