Stochastic Beams and Where to Find Them

Article Wouter Kool, Herke van Hoof, Max Welling
Presentation Dayana Savostianova

NRU HSE

26 09 2019

Applications

- Image captioning
- Neural machine translation
- Speech recognition

MATH

$I \sim \text{Categorical}(p_1, ..., p_n)$
$P(I = i) = p_i \quad \forall i \in N$
$\phi_i, i \in N$, $\exp \phi_i \propto p_i$

$$I \sim \text{Categorical}\left(\frac{\exp \phi_i}{\sum_{j \in N} \exp \phi_j}, i \in N\right). \tag{1}$$

The Gumbel distribution

$U \sim \text{Uniform}(0, 1)$
$G = \phi - \log(-\log U) : G \sim \text{Gumbel}(\phi)$
$G' = G + \phi' \sim \text{Gumbel}(\phi + \phi')$
Properties: we can *shift* Gumbel variables.

The Gumbel Max-Trick

The Gumbel-Max trick allows to sample from the categorical distribution by independently *perturbing* the log-probabilities ϕ_i with Gumbel noise and finding the largest element.

$G_i \sim \text{Gumbel}(0), i \in N$, $I^* = argmax_i\{\phi_i + G_i\}$

$I^* \sim \text{Categorical}(p_i, i \in N)$ with $p_i \propto \exp \phi_i$.

$G_{\phi_i} = G_i + \phi_i \sim \text{Gumbel}(\phi_i)$

$\forall B \subseteq N$:

$$\max_{i \in B} G_{\phi_i} \sim \text{Gumbel}\left(\log \sum_{j \in B} \exp \phi_j\right), \tag{2}$$

$$argmax_{i \in B} G_{\phi_i} \sim \text{Categorical}\left(\frac{\exp \phi_i}{\sum_{j \in B} \exp \phi_j}, i \in B\right). \tag{3}$$

Max and argmax are independent.

Difference: *ordered sample of size k without replacement.*

Theorem

For $k \leq n$, let $I_1^*, \ldots, I_k^* = \operatorname{argtopk} G_{\phi_i}$. Then $I_1^*, \ldots, I_k^*$ is an (ordered) sample without replacement from the $\text{Categorical}\left(\frac{\exp \phi_i}{\sum_{j \in N} \exp \phi_j}, i \in N\right)$ distribution, e.g. for a realization $i_1^*, \ldots, i_k^*$ it holds that

$$P\left(I_1^* = i_1^*, \ldots, I_k^* = i_k^*\right) = \prod_{j=1}^{k} \frac{\exp \phi_{i_j^*}}{\sum_{\ell \in N_j^*} \exp \phi_\ell} \tag{4}$$

where $N_j^* = N \setminus \{i_1^*, \ldots, i_{j-1}^*\}$ is the domain (without replacement) for the j-th sampled element.

Typically $p_{\boldsymbol{\theta}}(y_t|\boldsymbol{y}_{1:t-1})$ is defined as a softmax normalization of unnormalized log-probabilities $\phi_{\boldsymbol{\theta}}(y_t|\boldsymbol{y}_{1:t-1})$ with optional temperature T (default $T = 1$):

$$p_{\boldsymbol{\theta}}(y_t|\boldsymbol{y}_{1:l-1}) = \frac{\exp\left(\phi_{\boldsymbol{\theta}}(y_t|\boldsymbol{y}_{1:t-1})/T\right)}{\sum_{y'} \exp\left(\phi_{\boldsymbol{\theta}}(y'|\boldsymbol{y}_{1:t-1})/T\right)}. \tag{5}$$

The normalization is w.r.t. a single token, so the model is *locally normalized*. The total probability of a (partial) sequence $\boldsymbol{y}_{1:l}$ follows from the chain rule of probability:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:t}) &= p_{\boldsymbol{\theta}}(y_t|\boldsymbol{y}_{1:t-1}) \cdot p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:t-1}) \qquad (6) \\ &= \prod_{t'=1}^{t} p_{\boldsymbol{\theta}}(y_{t'}|\boldsymbol{y}_{1:t'-1}). \qquad (7) \end{aligned}$$

BEAM SEARCH

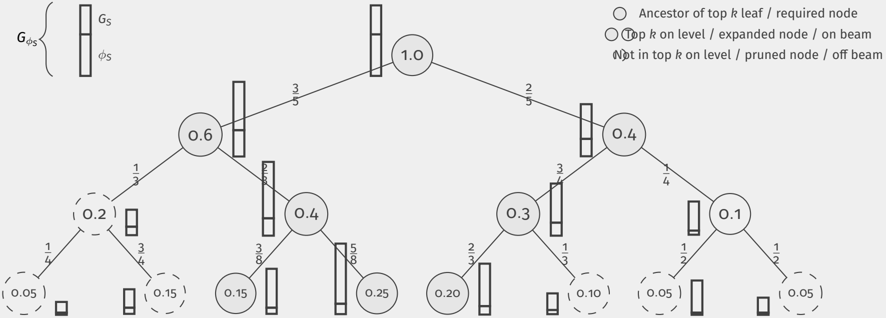
1. Limited-width breadth first search.
2. Often used as an approximation to finding the (single) sequence $\boldsymbol{y}$ that maximizes, or as a way to obtain a set of high-probability sequences from the model.
3. Expands at every step $t = 0, 1, 2, \ldots$ at most k partial sequences (those with highest probability) to compute the probabilities of sequences with length $t + 1$.
4. Terminates with a beam of k complete sequences.

The Gumbel-Top-k trick on a tree

We represent the sequence model as a tree.
Internal nodes at level t represent partial sequences $\boldsymbol{y}_{1:t}$
Leaf nodes represent completed sequences.
Normalized log-probability $\phi_i = \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^i)$.

- Compute $\phi_i = \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^i)$ for *all sequences* $\boldsymbol{y}^i, i \in N$. (Complete probability tree is instantiated)
- Sample $G_{\phi_i} \sim \text{Gumbel}(\phi_i)$, so G_{ϕ_i} can be seen as the perturbed log-probability of sequence $\boldsymbol{y}^i$.
- Let $i_1^*, ..., i_k^* = \arg\operatorname{top} k\, G_{\phi_i}$, then $\boldsymbol{y}^{i_1^*}, ..., \boldsymbol{y}^{i_k^*}$ is a sample of sequences without replacement.

Example on tree



Internal or leaf – the set S of leaves in the corresponding subtree, and $\boldsymbol{y}^S$ – the corresponding sequence.

$$\phi_S = \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^S) = \log \sum_{i \in S} \exp \phi_i. \tag{8}$$

$\forall S$ G_{ϕ_S} – maximum of the perturbed log-probabilities G_{ϕ_i} in the subtree leaves S.

$$G_{\phi_S} = \max_{i \in S} G_{\phi_i} \sim \mathrm{Gumbel}(\phi_S) \tag{9}$$

Gumbel noise $G_S \sim \mathrm{Gumbel}(0)$
$G_{\phi_S} = \phi_S + G_S$.

Children(S) – set of direct children of the node S Rule:

$$G_{\phi_S} = \max_{S' \in \text{Children}(S)} G_{\phi_{S'}}. \tag{10}$$

If we want to sample G_{ϕ_S} for all nodes, we can use the *bottom-up* sampling procedure: sample the leaves $G_{\phi_{\{i\}}} = G_{\phi_i}, i \in N$ and recursively compute G_{ϕ_S}. This is effectively sampling from the degenerate (constant) distribution resulting from conditioning on the children.

Top down

1. Initialize from the root
2. Rule: (from bottom up)
3. Sample the children conditionally on the parent variable (preserving max).
4. Sampling a set of Gumbels conditionally on their maximum being equal to a certain value can be done by first sampling the $\arg\max$ and then sampling the individual Gumbels conditionally on both the $\max$ and $\arg\max$.
5. For all leaves $(G_{\phi_{\{i\}}} =)\, G_{\phi_i} \sim \text{Gumbel}(\phi_i)$ is *independent*
6. The benefit of using top-down sampling is that if we are interested only in obtaining the top k leaves, we do *not* need to instantiate the complete tree.

1. Apply the top-down sampling procedure
2. At each level we only need to expand the k nodes with the highest perturbed log-probabilities G_{ϕ_S}
3. By the Gumbel-Top-k trick the result is a sample without replacement from the sequence model

We can also think of G_{ϕ_S} as the *stochastic score* of the partial sequence $\boldsymbol{y}^S$.

Why better?

1. A sampling procedure
2. Principled way to randomize a beam search.

Randomized beam search problems:

- Low-probability partial sequence need to be re-chosen, independently, again with low probability at each step. The result is a much lower probability to sample this sequence than assigned by the model. Intuitively, we should somehow *commit* to a sampling ‘decision’ made at step t.
- Stochastic Beam Search is better suited because it makes a *soft* commitment to a partial sequence by propagating the Gumbel perturbation of the log-probability consistently down the subtree.

Experiments

Different algorithms

- Beam Search
- Sampling
- Stochastic Beam Search
- Diverse Beam Search with G groups

Metrics

- Diversity – the fraction of unique *n*-grams in the *k* translations
- Quality of the sample – maximum BLEU score
- Diversity – mean BLEU score

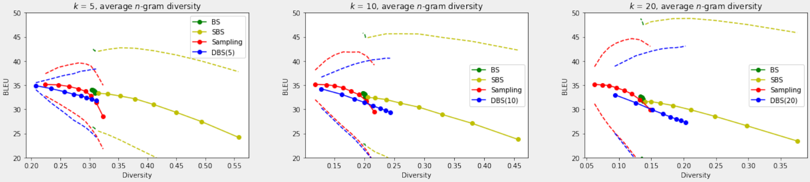
N-gram diversity

$$d_n = \frac{\text{\# of unique } n\text{-grams in } k \text{ translations}}{\text{total \# of } n\text{-grams in } k \text{ translations}}.$$

BLEU

Bilingual evaluation understudy
"The closer a machine translation is to a professional human translation, the better it is"

The wmt14.v2.en-fr.newstest2014 test set consisting of 3003 sentences



(a) $k = 5$ **(b)** $k = 10$ **(c)** $k = 20$

Figure: Minimum, mean and maximum BLEU score vs. diversity for different sample sizes k. Points indicate different temperatures/diversity strengths, from 0.1 (low diversity, left in graph) to 0.8 (high diversity, right in graph).

BLUE: $f(\boldsymbol{y}) = \text{BLEU}(\boldsymbol{y}, \boldsymbol{x})$
Entropy: $f(\boldsymbol{y}) = -\log p_\theta(\boldsymbol{y}|\boldsymbol{x})$

Evaluation methods

- Monte Carlo

$$\mathbb{E}_{\boldsymbol{y}\sim p_\theta(\boldsymbol{y}|\boldsymbol{x})}[f(\boldsymbol{y})] \approx \frac{1}{k}\sum_{i\in S} f(\boldsymbol{y}^i). \tag{11}$$

- StochasticBeamSearch (Normalized)

$$\mathbb{E}_{\boldsymbol{y}\sim p_\theta(\boldsymbol{y}|\boldsymbol{x})}[f(\boldsymbol{y})] \approx \sum_{i\in S} \frac{p_\theta(\boldsymbol{y}^i|\boldsymbol{x})}{q_{\theta,\kappa}(\boldsymbol{y}^i|\boldsymbol{x})} f(\boldsymbol{y}^i) \text{ if norm } (\frac{1}{W(S)}) \tag{12}$$

- Beam search (Norm)

$$\sum_{i\in S} p_\theta(\boldsymbol{y}^i|\boldsymbol{x}) f(\boldsymbol{y}^i) \text{ if norm } (\frac{1}{\sum_{i\in S} p_\theta(\boldsymbol{y}^i|\boldsymbol{x})}) \tag{13}$$

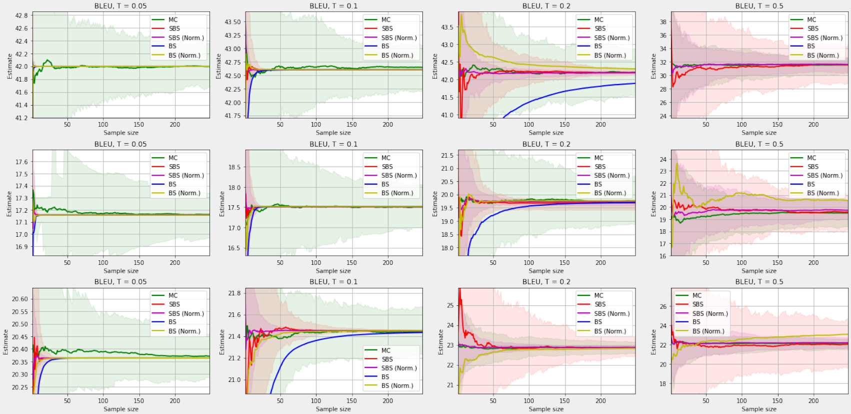


Figure: BLEU score estimates for three sentences sampled/decoded by different estimators for different temperatures.

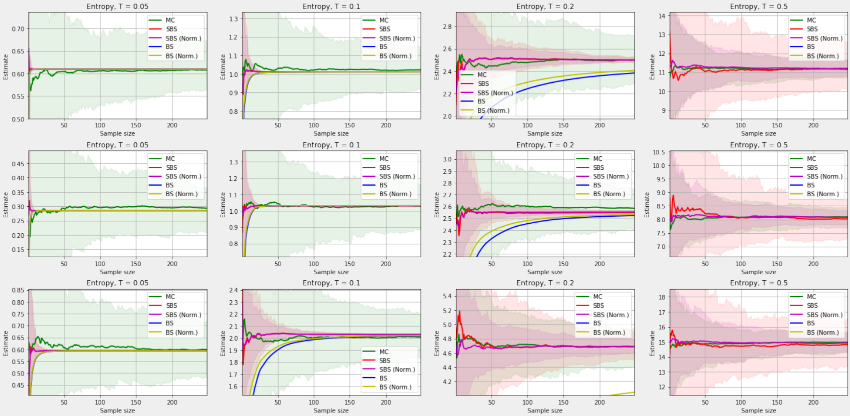


Figure: Entropy score estimates for three sentences sampled/decoded by different estimators for different temperatures.

Stochastic Beam Search shares some of the benefits of these heuristic variants, such as the ability to control diversity or produce randomized output.

Benefits

- Linear towards k and the length of the sequence
- Ability to control diversity
- Produce randomized output
- Easy to implement on top of a beam search as a way to sample sequences without replacement

Thank you for your **attention**!

Algorithm 1 StochasticBeamSearch($p_{\boldsymbol{\theta}}$, k)

Input: one-step probability distribution $p_{\boldsymbol{\theta}}$, beam/sample size k
Initialize BEAM empty
add $(\boldsymbol{y}^N = \varnothing, \phi_N = 0, G_{\phi_N} = 0)$ to BEAM
for $t = 1, \dots,$ steps **do**
  Initialize EXPANSIONS empty
  for $(\boldsymbol{y}^S, \phi_S, G_{\phi_S}) \in$ BEAM **do**
    $Z \leftarrow -\infty$
    for $S' \in \text{Children}(S)$ **do**
      $\phi_{S'} \leftarrow \phi_S + \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^{S'} | \boldsymbol{y}^S)$
      $G_{\phi_{S'}} \sim \text{Gumbel}(\phi_{S'})$
      $Z \leftarrow \max(Z, G_{\phi_{S'}})$
    end for
    for $S' \in \text{Children}(S)$ **do**
      $\tilde{G}_{\phi_{S'}} \leftarrow -\log(\exp(-G_{\phi_S}) - \exp(-Z) + \exp(-G_{\phi_{S'}}))$
      add $(\boldsymbol{y}^{S'}, \phi_{S'}, \tilde{G}_{\phi_{S'}})$ to EXPANSIONS
    end for
  end for
  BEAM $\leftarrow$ take top k of EXPANSIONS according to $\tilde{G}$
end for
Return BEAM