Breaking the Beam Search Curse: A Study of (Re-)Scoring Methods and Stopping Criteria for Neural Machine Translation

Oregon State University

Yilin Yang[1] Liang Huang[1,2] Mingbo Ma[1,2]

[1] School of EECS, Oregon State University [2] Baidu Research, USA

Baidu Research

BACKGROUND: BEAM SEARCH ALGORITHM

While decoding with a sequence-to-sequence model, we could not afford to search globally for optimal output sequence, so researchers often resort to beam search algorithm to approximate exact search. The beam search algorithm expands B_{t-1} to B_t as follows:

$$B_0 = [\langle \texttt{<s>},\ p(\texttt{<s>} \mid \mathbf{x})\rangle]$$

$$B_t = \overset{b}{\mathbf{top}}\{\langle \mathbf{y}' \circ y_t,\ s \cdot p(y_t|\mathbf{x}, \mathbf{y})\rangle \mid \langle \mathbf{y}', s\rangle \in B_{t-1}\}$$

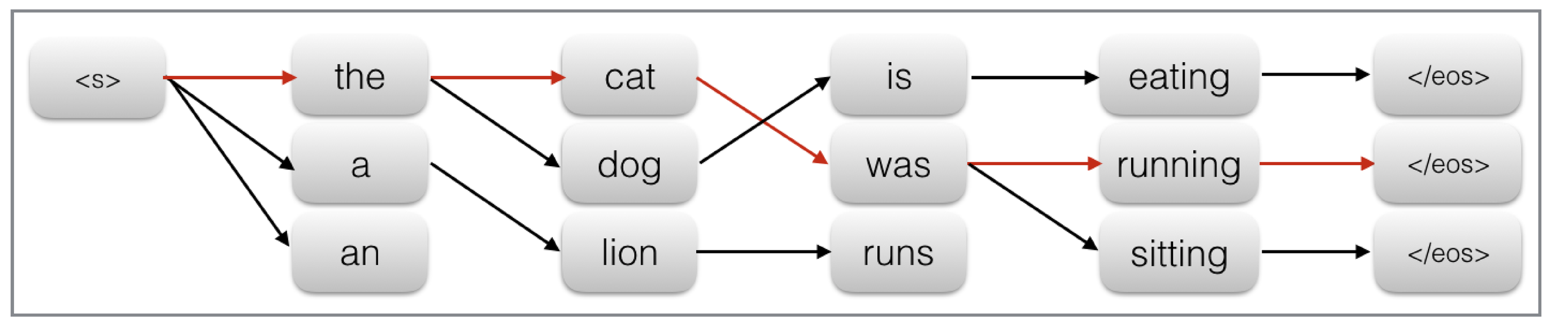

Figure 1: Examples of beam search algorithm with beam size 3. Red arrows denote greedy search (beam size 1).

In the end, the algorithm chooses the candidate with highest log-probability:

$$\mathbf{y}^* = \underset{\mathbf{y}:comp(\mathbf{y})}{\mathbf{argmax}}\ sc(\mathbf{x}, \mathbf{y}) = \underset{\mathbf{y}:comp(\mathbf{y})}{\mathbf{argmax}} \sum_{t \le |\mathbf{y}|} \log p(y_t \mid \mathbf{x}, \mathbf{y}_{<t})$$

where $comp(\mathbf{y}) \triangleq (\mathbf{y}_{|\mathbf{y}|} = \texttt{</eos>})$ returns the completeness of a hypothesis.

BEAM SEARCH CURSE

It's widely observed that as beam size increases after 5, the performance of sequence-to-sequence models, as quantified by the BLEU score, drops greatly. Since the models could not leverage the computational power from wider beams, we call this phenomenon the *Beam Search Curse*.

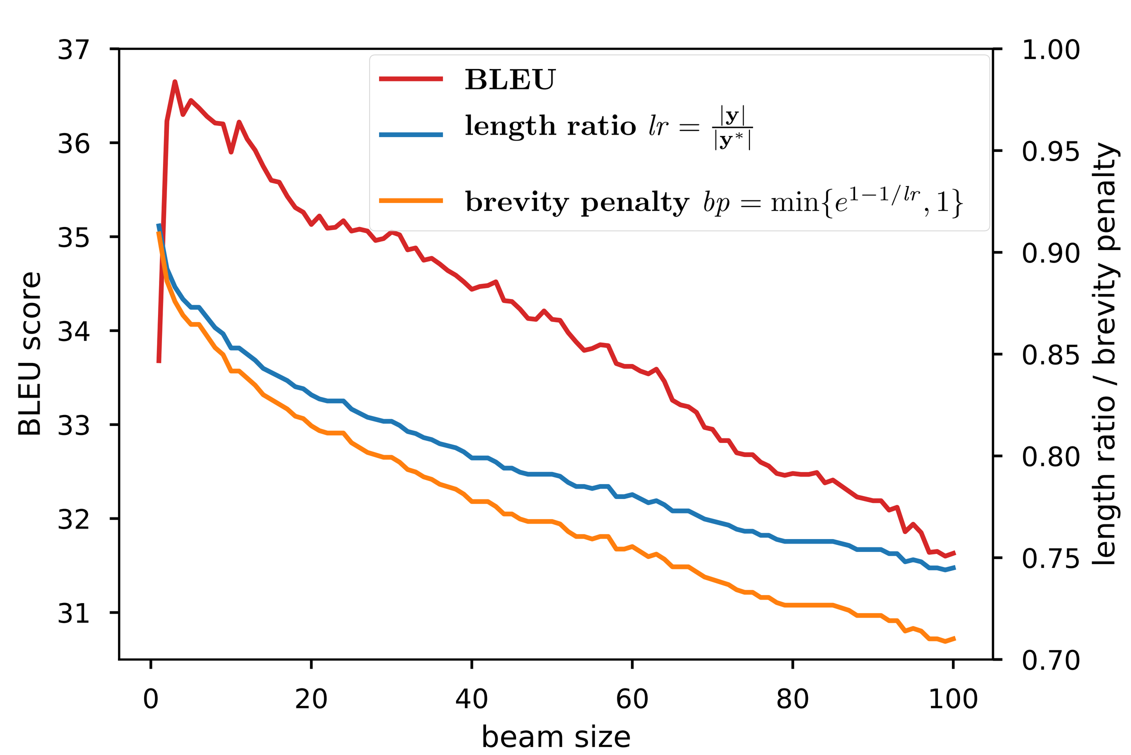

Figure 2: While the BLEU score drops with an increasing beam size (after 5), the brevity penalty drops with a similiar curve.

EXPERIMENTAL SETUP

1. Based on OpenNMT-py, a PyTorch reimplementation of Torch-based OpenNMT (Klein et al., 2017).
2. 2M Chinese-English sentence pairs for training.
3. Used byte-pair encoding (BPE) (Senrich et al., 2015) to reduce vocabulary sizes down to 18k/10k respectively.
4. Chinese to English: NIST 06 newswire portion (616 sentences) for dev; NIST 08 newswire portion (691 sentences) for test.

WHY THE CURSE EXISTS

1. As beam size increases, the more candidates it would explore. Therefore, it becomes easier to find the </eos> symbol and terminate. Left figure shows that the </eos> indices decrease steadily with wider beams.
2. Then, because of the internal property of log-probability, shorter candidates have clear advantages *w.r.t.* model score.

As a conclusion, the search algorithm would find shorter candidates, and prefer even shorter ones among them.

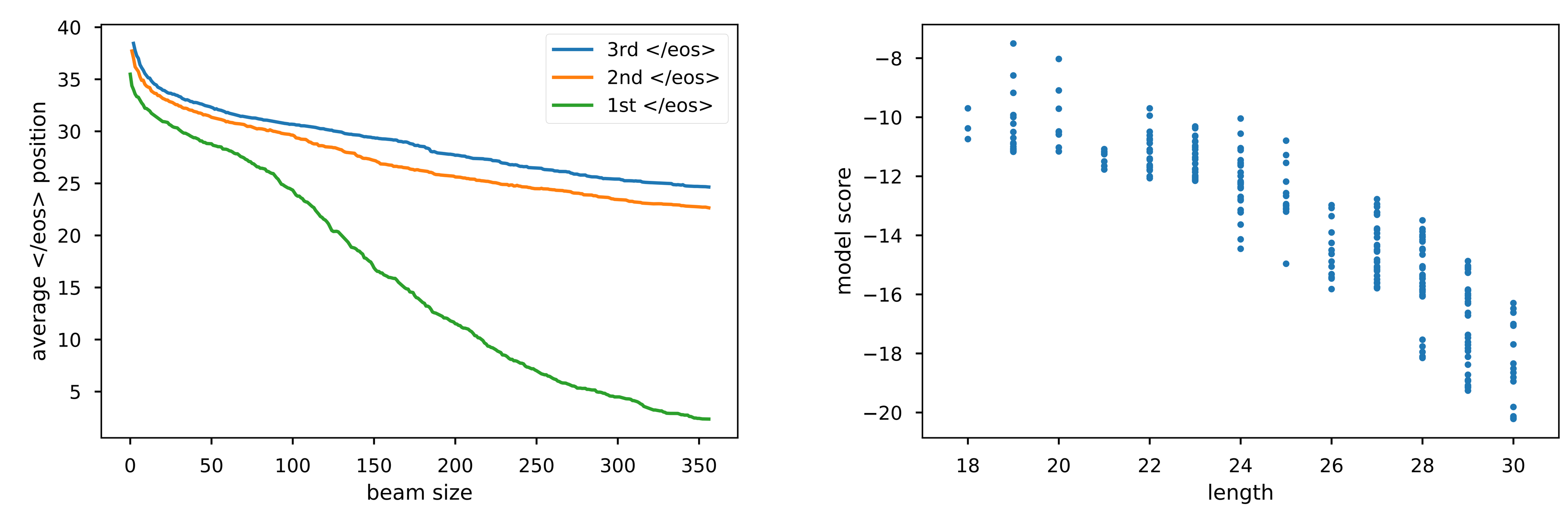

Figure 3: Left: Searching algorithm with wider beams generates </eos> earlier. Right: The model score (log-probability) strongly prefers shorter candidates.

HOW TO BREAK THE CURSE

Previous Methods

1. Length Normalization (Bahdanau et al., 2014): normalize the score by its length.
2. Word-Reward (He et al., 2016): add reward r to each word.
3. Bounded Word-Reward (Liang et al., 2017): add reward r to each word up to a bound.

Rescoring with Length Prediction We use a 2-layer MLP, which takes the mean of source hidden states as input, to predict the generation ratio $gr(\mathbf{x})$. Then we can get our predicted length $L_{pred}(\mathbf{x}) = gr(\mathbf{x}) \cdot |\mathbf{x}|$.

Bounded Word-Reward w/ Predicted Length To favor longer generation, we add rewards r to each word up to its predicted length.

$$L(\mathbf{x}, \mathbf{y}) = \min\{|\mathbf{y}|, L_{pred}(\mathbf{x})\} \qquad \hat{sc}(\mathbf{x}, \mathbf{y}) = sc(\mathbf{x}, \mathbf{y}) + r \cdot L(\mathbf{x}, \mathbf{y})$$

where $sc(\mathbf{x}, \mathbf{y})$ is the original model score (log-probability).

Bounded Adaptive-Reward Instead of a tuned reward r, we add an adaptive reward to each step based off local beam information. With beam size b, the reward for time step t is the average negative log-probability of the words in the current beam.

$$r_t = -(1/b)\sum_{i=1}^{b} \log p(\text{word}_i) \qquad \hat{sc}(\mathbf{x}, \mathbf{y}) = sc(\mathbf{x}, \mathbf{y}) + \sum_{t=1}^{L(\mathbf{x}, \mathbf{y})} r_t$$

BP-Norm Instead of adding rewards, we apply brevity penalty to the length-normalized model score.

$$bp = \min\{e^{1-1/lr}, 1\} \qquad \hat{sc}(\mathbf{x}, \mathbf{y}) = \log bp + sc(\mathbf{x}, \mathbf{y})/|\mathbf{y}|$$

DISCUSSION

Among all methods, we recommend **BP-Norm** for the following reasons:

1. BP-Norm works equally well with others, while doesn't contain any hyper-parameters.
2. BP-Norm is intuitive and in the same form as BLEU. Both of their exponential forms are products of brevity penalty term and geometric mean of probabilities (BP-Norm) or accuracies (BLEU).

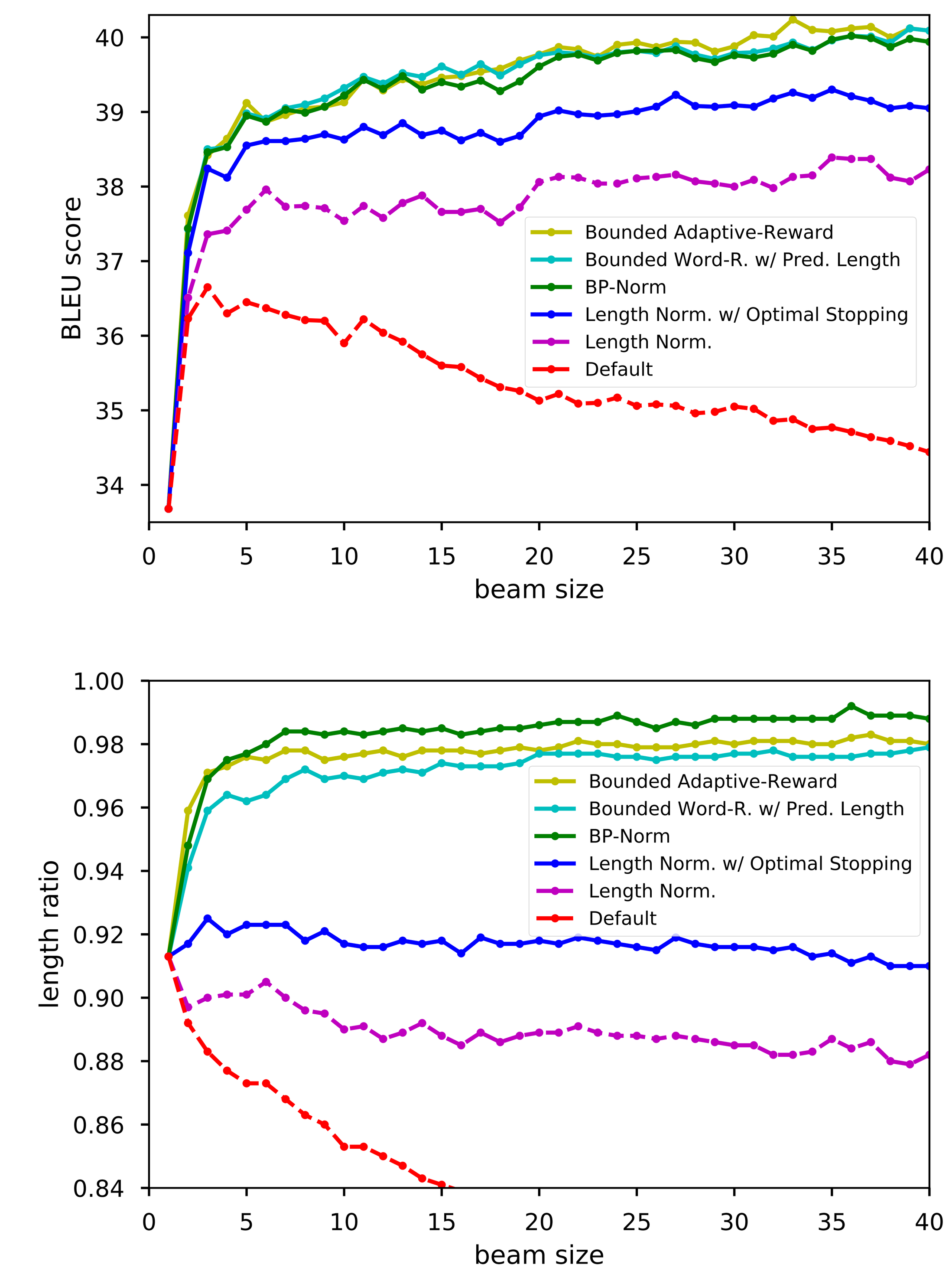

Figure 4: BLEU and length ratios of various rescoring methods.

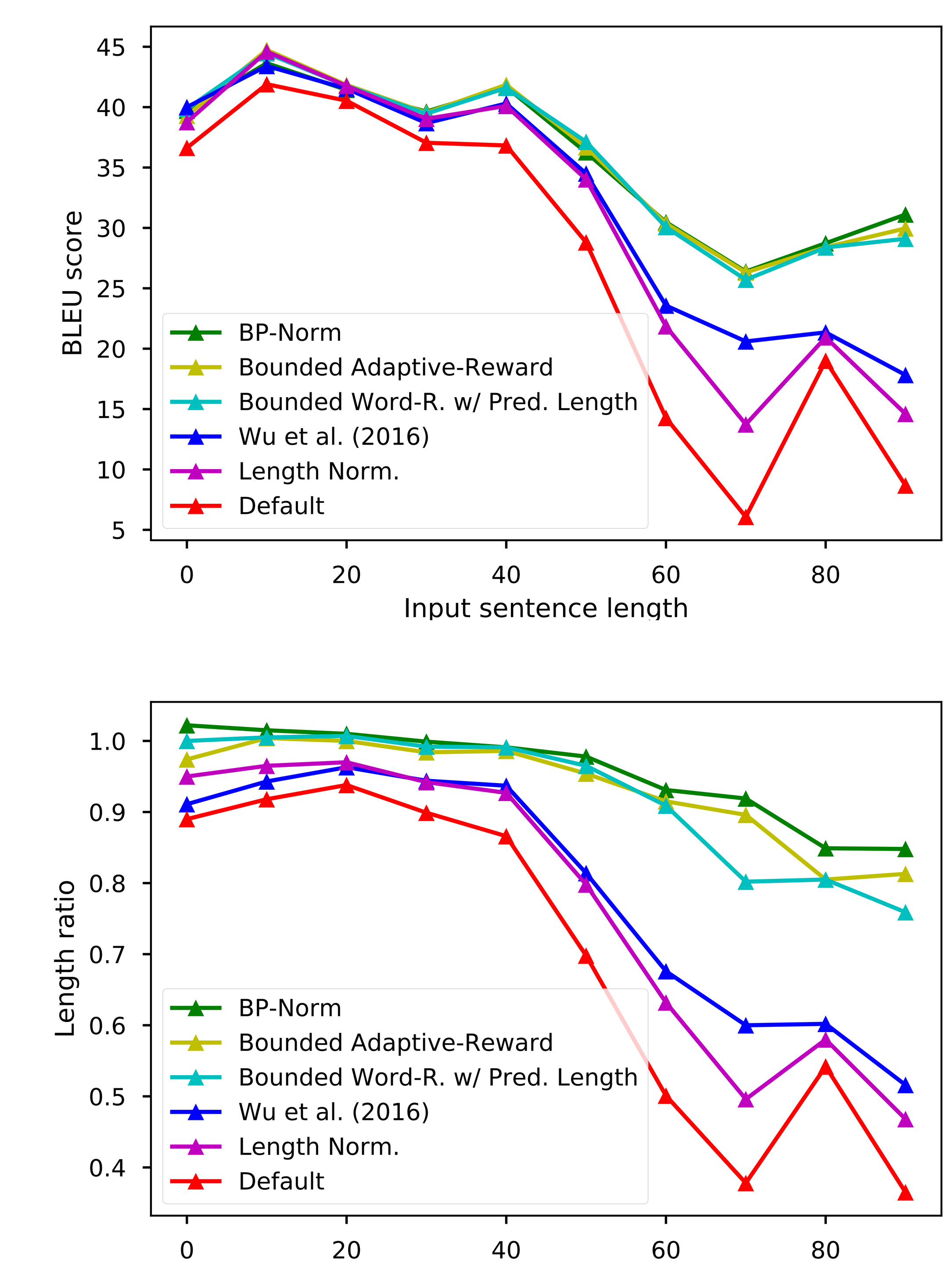

Figure 5: BLEU and length ratios over various input sentence lengths.