Improving Beam Search by Removing Monotonic Constraint for Neural Machine Translation

Raphael Shu and **Hideki Nakayama**
The University of Tokyo

Overview

- Neural machine translation models rely on Beam Search to produce output words.
- Beam search has to sacrifice an existing decoding path to explore a new candidate.
- As the output sentence is decoded in left-to-right order, once a decoding path is discarded, it can not be recovered again. We call it **monotonic constraint**.
- In this work, we extend beam search to allow revisiting a discarded decoding path in the past.
- The algorithm is implemented with a single priority queue,
  - we refer it to as **single-queue decoding (SQD)**.

Proposed Algorithm

Algorithm 1 Single-queue decoding

for $t \leftarrow 1$ to T **do**
  $S \leftarrow$ pop top-B hypotheses
  $S' \leftarrow$ expand S to get $B \times K$ new hyps
  Evaluate scores of hyps in S' with score(y)
  Push S' into H
  if $\#(\text{finished hyps in } H) \geq B$ **then**
    break
$\hat{y} \leftarrow$ best finished hyp in H
output $\hat{y}$

priority queue H

2.3	<s> there is an
2.3	<s> an apple
2.2	<s> there is a
1.9	<s> there was
. . .	

Score Function

- The score function adds two auxiliary penalties to log likelihood:

$$\text{score}(y) = \frac{1}{|y|^{\lambda}} \log p(y|X) + \alpha\,\text{PG}(y) + \beta\,\text{LMP}(y)$$

progress penalty — length matching penalty

$$\text{PG}(y) = \begin{cases} 0 & \text{if } y \text{ finished} \\ \frac{|y|^{\gamma}}{|X|^{\gamma}} & \text{otherwise} \end{cases} \qquad \text{LMP}(y) = \begin{cases} 0 & \text{if } y \text{ finished} \\ \mathbf{I}(\text{H}(p_x^l, p_y^l) > \tau) & \text{otherwise} \end{cases}$$

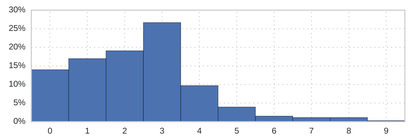
cross-entropy of two Gaussians

$$\text{H}(p_x^l, p_y^l) = \frac{1}{2}\log(2\pi{\sigma_y}^2) + \frac{{\sigma_x}^2 + (\mu_y - \mu_x)^2}{2{\sigma_y}^2}$$

- The **progress penalty** (PG) encourage the algorithm to select longer candidates.
- The **length matching penalty** (LMP) punishes the hypotheses that tend to produce translations much longer or shorter than expectation.
  - p_x^l : the Gaussian of expected output length given x
  - p_y^l : the Gaussian of expected output length if continue to decode current hypothesis

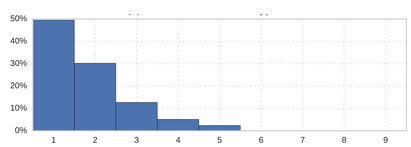
Analysis

- How many discarded hypotheses are recovered when translating one sentence?



When translating each sentence, the algorithm generally revisits no more than 5 hypotheses, so the decoding speed is not significantly worsen.

- How many steps does the algorithm look back to recover a hypothesis?

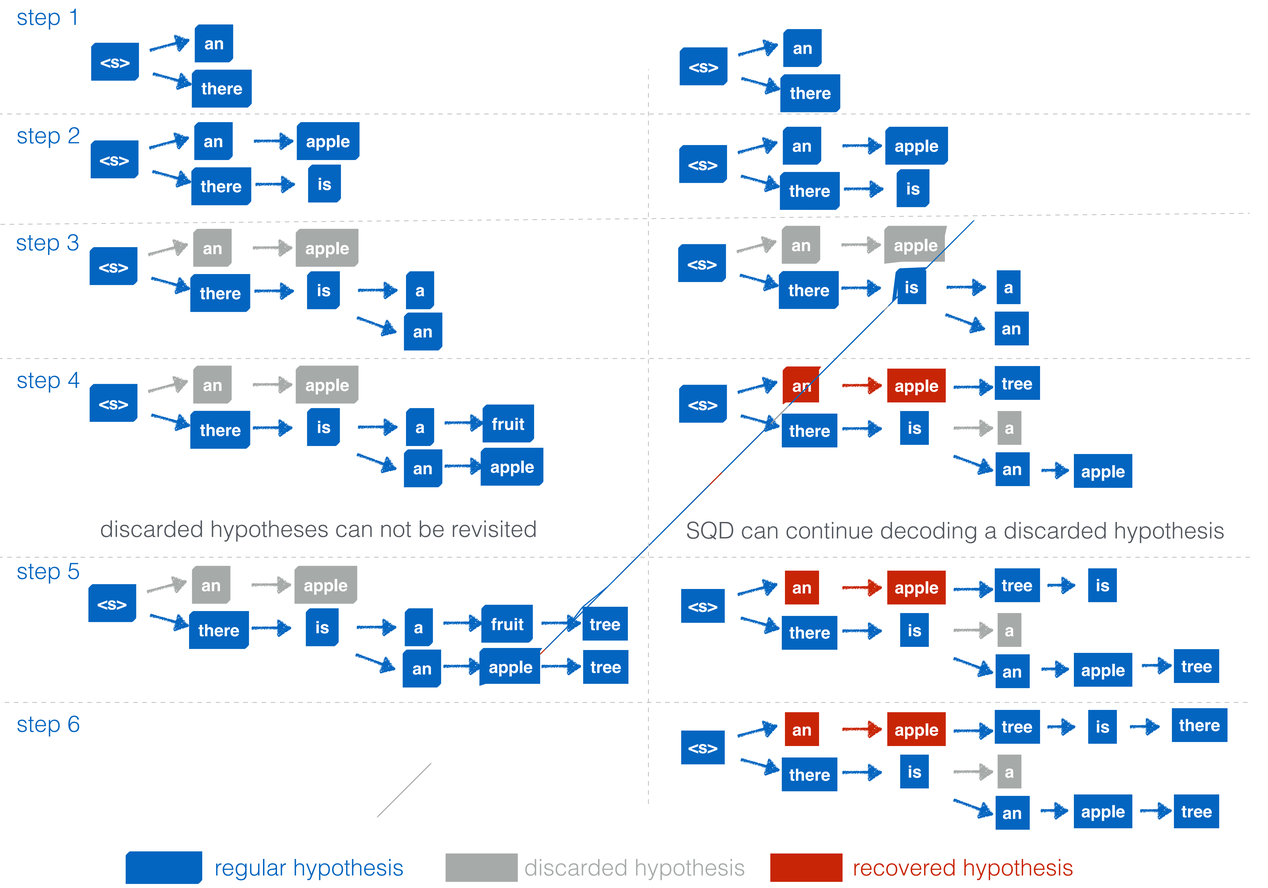


The algorithm generally recover hypotheses that are discarded no more than 3 steps ago.

- A comparison of decoding speed

algorithm	speed
beam search	208 ms/sent
SQD w/ PG	225 ms/sent
SQD w/ PG, LMP	260 ms/sent

Compare **beam search** with **SQD** (beam size = 2)



Quantitive Results

	Test BLEU(%)			
	BS=3	BS=5	BS=8	BS=12
beam search w/ LN	37.69	37.93	38.26	38.38
SQD w/ PG	38.18	38.68	**38.98**	**39.02**
SQD w/ PG, LMP	**38.37**	**38.73**	38.89	38.98

Results on English-Japanese translation task

Insights

- SQD algorithm generally improves the translation results.
- Length matching penalty (LMP) is only effective when a small beam size is used.
- Without PG, the translation performance hurts (not shown in the table).

Have Fun

- Contact me: shu@nlab.ci.i.u-tokyo.ac.jp
- Deep Learning Monitor: https://deeplearn.org

東京大学 THE UNIVERSITY OF TOKYO
中山英樹研究室 NAKAYAMA LABORATORY

RI BL

text

test

text

text

[Shu et al., 2015]

Text

123

Text

Text