

Beam Search Strategies for Neural Machine Translation

1 Main Idea

The standard beam search strategy for Neural Machine Translation (NMT) is for the decoder to predict the target sequence word-by-word and maintain a fixed amount of potential word candidates to predict at each step.

The drawbacks of this approach are that it is less adaptive, because:

- Some candidates might not be as good as the current best
- Good candidates might not be considered because they missed out on the threshold for candidate inclusion marginally. Addressing this by naively increasing beam search size will result in slowing down the decoder.

$$beam_size \propto model_accuracy$$

$$beam_size \propto \frac{1}{model_performance}$$

The paper proposes a more flexible decoder strategy, by pruning the search graph, and reducing the number of candidates with the same partial hypothesis (shared past).

2 Background

Standard beam search builds a translation from left-to-right and keeps a fixed number (beam) of translation candidates with the highest log-probability at each time step. If an end-of-sequence (EOS) token is encountered, we reduce the beam size by 1 and add the sequence to the candidate list of sequences. When the beam size eventually becomes 0, we evaluate the log-probability of all the sequences weighted by sequence length and pick the translation with the highest log-probability score.

3 Method

4 separate beam search strategies are proposed. One or more of these can be applied on top of vanilla beam search.

- Relative threshold pruning: This eliminates candidates are beyond a relative distance to the current best candidate.

$$score(cand) \leq rp * \max_{c \in C} score(c)$$

where rp is the pruning threshold.

- Absolute threshold pruning: Almost the same as the previous strategy except that the difference between the best and pruned candidates is determined by an Absolute value ap .

$$score(cand) \leq \max_{c \in C} score(c) - ap$$

- Relative local threshold pruning: This is similar to the first strategy, but instead of considering the log-probability of the entire sequence, only the log-probability of the last generated word instead of the total sequence score.

$$score_w(cand) \leq rpl * \max_{c \in C} score_w(c)$$

- Maximum candidates per node: This strategy eliminates candidates based on whether too many current candidates share the same predecessor words, in an attempt to diversify the predictions.

$$cand_{pred=P} \text{ if } \sum_{c_{pred=P}} 1 > mc$$

where mc is the maximum candidate threshold for candidates that share common predecessor words.

4 Observations

•

References