Beam Search Strategies for Neural Machine Translation

1 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 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

The drawbacks of a vanilla beam-search is 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.

$$beam\_size \propto model\_accuracy$$

$$beam\_size \propto \frac{1}{model\_performance}$$

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, the beam-size is reduced by 1 and the sequence is added to the candidate list of translations. When the beam-size eventually becomes 0, the log-probability of all the sequences weighted by sequence length is evaluated and the translation with the highest log-probability score is picked.

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 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.

English → German translation done at the sub-word level (reduces the computational complexity of the output softmax). English → Chinese translation done at the word level.

NMT implementation similar to the Neural Attention [1] paper. Embedding dimension of 620 used, with RNN GRU unit for the latent representation of 1000 units. SGD is the learning algorithm used. Number of epochs not discussed, but a batch-size of 64 is used for training.

Experiments done to choose the most conservative threshold that doesn't degrade translation accuracy.

Beam-sizes of 5 to 14 are experimented with.

4 Observations

- Relative pruning worked best for beam-size 5, and absolute pruning worked best for beam-size 14.
- For English → German, speed-up of 13% for beam-size 5, and 43% for beam-size 14.
- For English → Chinese, speed-up of 10% for beam-size 5, and 24% for beam-size 14.
- Adding diversity to the decoder using the Maximum Candidates pruning strategy did not improve translation results.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.