Summary
Effective Approaches to Attention-based Neural Machine Translation
Author: Shobhit Raj Gautam (2019201056)

1. Introduction

NMT is a large neural network that is trained in an end to end fashion for translating one language into another. The model produces 1 word at a time as there is no large language model stored with small memory footprint. The attention model is classified in 2 approaches: a global approach in which all source words are attended and a local one whereby only a subset of source words are considered at a time.

2. Nueral Machine Translation

NMT basically has 2 components: An encoder with which computes a representation S for each source sentence. An decoder which generates translation one word. One can use probability of decoding each word based on softmaxing the transformation function g(x) that outputs a vocabulary-sized vector based on input h_j which is hidden RNN unit calculated based on previous hidden RNN unit.

3. Attention-based Model

The 2 Approaches: global and local. Both take hidden state at time t called h_t at top layer of stacking LSTM to produce context vector c(t) to produce current target word y(t). The attentional hidden layer h(bar)(t) is combo of both h(t) and c(t) that is used to produce predictive distribution using softmax. i.e The attentional vector is transformed using the softmax layer to produce the predictive distribution. We are using the softmax layer as we have to found the most probable word from all the available words in our vocabulary.

- Global Attention: Global attention takes into consideration all encoder hidden states to derive the context vector (c(t)). In order to calculate c(t), we compute a(t) which is a variable length alignment vector. The alignment vector is derived by computing a similarity measure between h(t) and h_bar(s) where h(t) is the source hidden state while h_bar(s) is the target hidden state.
  Drawback: It focus on all source side words for all target words, it is computationally very expensive and is impractical when translating for long sentences. To overcome this deficiency local attention chooses to focus only on a small subset of the hidden states of the encoder per target word.

- Local Attention: The model first generates an aligned position p(t) for each target word at time t. In contrast to the global attention model where we assume monotonic alignment, we learn aligned positions in local attention. In other words apart from learning translations you also learn if the order of translation is different from the source sentence. The context vector (c(t)) is derived as a weighted average over the set of source hidden states within the window [p(t) — D, p(t)+D]; D is empirically selected. As compared to the global alignment vector local alignment vector a(t) is now fixed dimensional.
  Types of Local attention:
  1. Monotonic Alignment(local-m):
  Set p(t)=t, which means that we are assuming that source and target sequences are roughly monotonically aligned. Alignment vector is the same as the global alignment.
  2.Predictive alignment (local-p):
  Our model predicts an aligned position by multiplying S with sigmoid which is product of the model parameters which will be learned to predict positions. S is the source sentence length. To favor alignment points near p(t) , we place a Gaussian distribution centered around p(t).

Global attention is computationally more expensive and is useless for long sentences while local attention focusses on D hidden states on both sides of p(t) to overcome this.
Local attention has 2 flavors local-m in which the source and target alignment are assumed to be the same and local-p where we calculate the p(t).

4.Input-Feeding Format:
The attention decisions are made independently which means previously predicted alignments does not influence next alignment. So for future alignments h_bar(t) is concatenated with input to use past information. This is done to make model fully aware of previous alignment choices and create a very deep network spanning both horizontally and vertically. Also it can be applied to general stacking recurrent architectures, including non-attentional models thus making it flexible and suitable

5. Experiments:
The effectiveness of model uses english and german in both directions and performances are reported in case sensitive BLEU. The translation quality using two types of BLEU: tokenized BLEU to be comparable with existing NMT work and NIST BLEU to be comparable with WMT results. The code runs on MATLAB with GPU Tesla K40 with running for around 10 days.

- Improvements: By reversing source sentence , using dropout, the global attention approach ,using the input-feeding approach, local attention model with predictive alignment, using a better alignment function, the content-based dot product one with dropout.
  All these gave significant boost.

6. Analysis:

- Non-attentional model with dropout is most robust as it minimizes test error but initially learns slower.
- This model is more effective in translating long sentences and quality remains as sentence size increases and even performs better than all other models.
- The local attention model with predictive alignments (local-p) is best in terms of BLEU and in case of dot, global attention is best.
- The Alignment Quality is given by AER metric and global has better AER than local attention.
- Also results from sample shows attentional model is better than non attention model when it comes to translating long sentences.

7. Conclusion:

This paper proposes two attention approaches each with their own benefit. The global approach which always looks at all source positions and the local one that only attends to a subset of source positions at a time.The experiments are done on our models in the WMT translation tasks between English and German in both directions. Along with that compared different alignment functions and choose best performer and attention-based NMT models are superior to nonattentional ones.