Attention Summary and Results

Jeremy Irvin

November 14, 2016

Summary: In their paper *Neural Machine Translation By Jointly Learning to Align and Translate*, Bengio et. al. propose an attention mechanism as a supplement to a neural machine translation (NMT) model which allows the model to perform a soft-search over the source sentence for relevant words in predicting the target translation. The previous model (proposed by Sutskever et. al) uses a deep (multilayer) LSTM encoder to first translate a variable-length sequence of words in the source language into a fixed-length vector, and another deep LSTM decoder to then translate the fixed-length vector into a variable-length sequence of words in the target language. These models are trained end-to-end (as well as the embeddings of the words in both the source and target languages) in order to maximize the probability of a correct translation given a source sentence.
The most novel addition to this NMT model is an attention mechanism. More explicitly, the hidden state output by the decoder at the tth timestep is now a function of the previous hidden state, previous true word (or predicted word during test time), *and* a context vector:

$$s_t = f(s_{t-1}, y_{t-1}, c_t).$$

The context vector at the tth timestep c_t is a linear combination of the hidden states h_s output by the encoder - one hidden state for each source word (the final hidden state if the LSTM encoder is multilayer; the number of hidden states M_x depends on the input x):

$$c_t = \sum_{s=1}^{M_x} \alpha_{ts} h_s$$

The coefficients α_{tj} of this linear combination are computed by taking a softmax over a (scoring) function of the source hidden states h_s and the state output by the decoder at timestep t, $\hat{h}^t$:

$$\alpha_{ts} = \frac{\exp(\text{score}(\hat{h}_t, h_s))}{\sum_{s'} \exp(\text{score}(\hat{h}_t, h_{s'}))}$$

where the score function is one of (these were the scoring functions tested by Manning et. al)

$$\text{score}(\hat{h}_t, h_s) = \begin{cases} \hat{h}_t^T h_s & \textit{dot} \\ \hat{h}_t^T W_\alpha h_s & \textit{general} \\ v_\alpha^T \tanh(W_\alpha[\hat{h}_t; h_s]) & \textit{concat} \end{cases}$$

Finally an attentional hidden state is then produced by

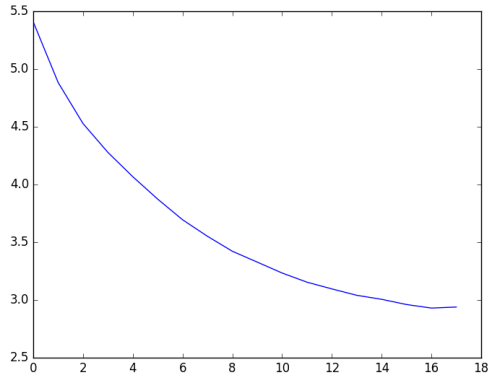
$$\tilde{h}_t = \tanh(W_c[c_t; \hat{h}_t])$$

which is fed into a softmax over the target vocabulary to model the conditional probability of the target word at time t given the previous words in the target sequence $y_{<t}$ and the input sequence x:

$$p(y_t | y_{<t}, x) = \text{softmax}(W_s \tilde{h}_t)$$

where W_s is a projection from the hidden dimension into the target vocabulary.
By allowing the model to weight the source hidden states and use this information to make predictions, attention mechanisms can essentially learn to focus on certain words in the input when learning to translate. This addition improved BLEU score to get close and beat state of the art at the time in several translation tasks (such as English to French and English to German).

Experiments:
I ran several experiments on the English to Vietnamese (133K sentence pairs, 50K vocab) and English to German (4.5M sentence pairs, 50K frequent words) datasets. Most of my initial experiments used the *concat* method proposed in the paper, which was unfortunately proven to be ineffective for every single one of my experiments (the network could overfit but validation and test errors were high). I tested ranges of sizes from 64 to 256 embedding dimensions, 128 to 512 hidden dimensions, batch sizes of 4 to 128, number of layers in the encoder/decoder from 1 to 4. These results coincide with the results from Manning et. al. which also state that the content-based function *concat* for scoring is ineffective.
My most recent experiments consist of using the *dot* method as the content-based function for scoring on the Vietnamese corpus only due to much shorter training times. I split the corpus into a training set of 120000 and validation set of 10000. I tested models with embedding dimension 256, hidden dimension 512, number of layers from 1 to 2, with and without early stopping using cross validation, with and without dropout, and with and without reversing the source. My results coincide almost exactly with the results of the paper - the models which performed best reversed the source sentence, used dropout and early stopping for regularization. The scores are difficult to compare directly due to the use of different languages (and thus very different grammatical structure as well as corpus size) However, I am able to achieve 18.5 BLEU score on the 2013 test set. Moreover, the model which achieves this used 256 embedding dimension, 512 hidden dimension, a single layer LSTM, 0.2 dropout, early stopping (if the validation loss decreases by 0.1 or less between two epochs), source sentence reversal, and 30 max length cutoff (data pairs which had source or target sentence longer than 30 pairs were thrown out at train time and at test time). Here are the training and validation loss curves for the best model, results of test examples not in the corpus with their alignment histograms from the best model, as well as alternate experiment results:



Example 1: The scientists used to like the research but now they are no longer interested .
Best model output: `Những sinh vật quen thuộc với những nghiên cứu mà chúng không còn quan tâm hơn .`
Google translation: These creatures are familiar with the research that we no longer care about more.
The model does a decent job with this sentence, as exemplified by the clear alignment scores (for example, words near the beginning of the source are clearly aligned with words near the beginning of the

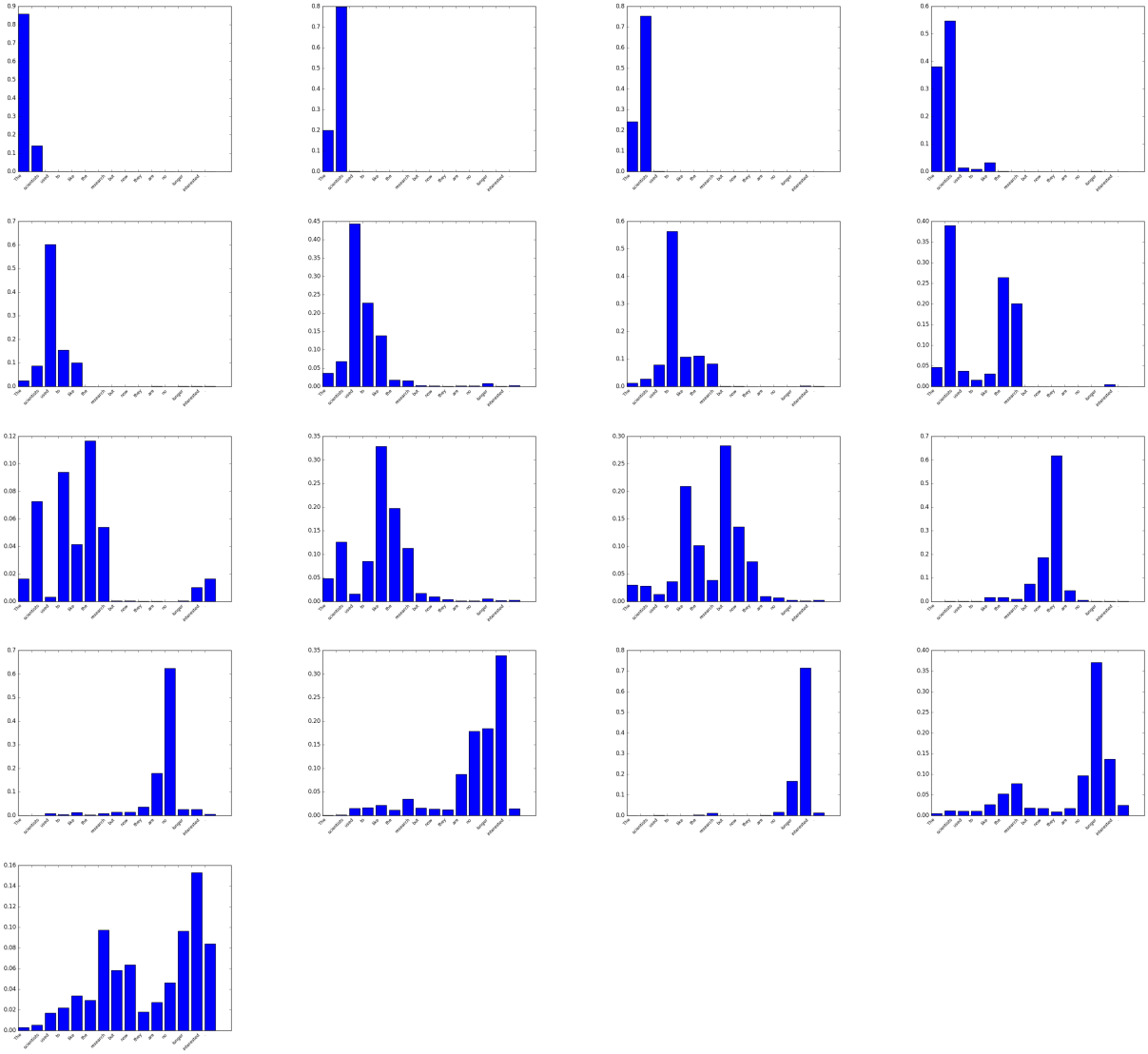


Figure 1: Alignment distributions for the output sentence in example1. Each graph corresponds to the word in the output sentence (ordered left right top down).

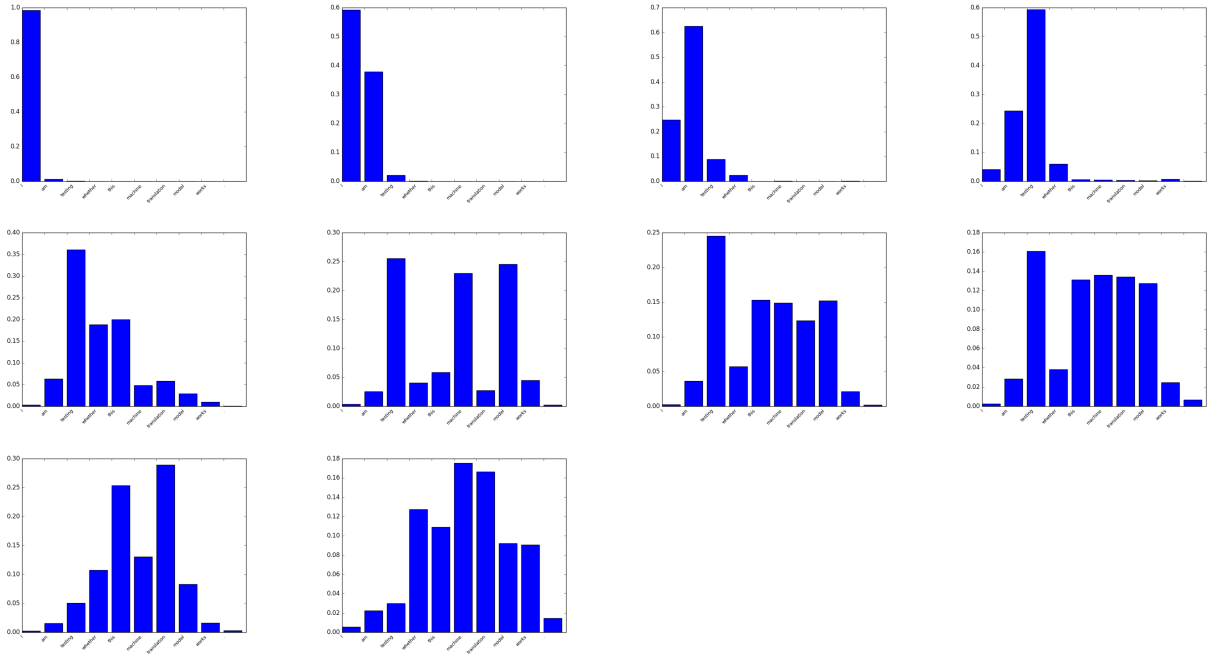


Figure 2: Alignment distributions for the output sentence in example2. Each graph corresponds to the word in the output sentence (ordered left right top down).

source, and similarly for the middle and end).

Example 2: I am testing whether this machine translation model works .
Best model output: `Tôi đã thử nghiệm một mẫu hình mẫu này .`
Google translation: I tested a sample of this pattern. The model does a poorer job with this translation, as exemplified by the higher entropy distributions in the above graphs. This sentence is much different than those found in the corpus, so this is reasonable.