Paper: *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*

Nirajan Koirala

CSC 9010_001

Summary

Neural Machine Translation is an end-to-end automated machine translation system which is better than conventional phrase-based translation systems in many ways. Although better than conventional translation systems, NMTs suffer with many issues like lack of robustness and authors have presented Google's Neural Machine Translation System in this paper which attempts to tackle many of the issues that NMTs suffer. GNMTs consists of deep LSTM networks and use residual connections to address the issue of gradient vanishing/exploding problems. These systems also have higher translation speeds and handle rare words more robustly using sub-word units. For the system to cover all the words in the source sentence during translation, the beam search technique is employed with length-normalization and they have also refined the models using reinforcement learning. More importantly, their implementation works even better on production data and GNMTs are able to reduce the translation errors by 60% when compared to conventional translation systems.

In section 2, authors have provided a history of NMT systems and how many techniques were proposed to improve them with different techniques like attention mechanisms and sentence-level loss minimization. A description of their current model's architecture and its workings is provided in the next section and their model follows the common sequence-to-sequence learning framework with attention and has 3 components: an encoder network, a decoder network and an attention network. In their experiments they found that depth of NMT systems is really important to capture subtle irregularities in the source and target languages but simply adding more layers only work up-to a certain number of layers and the network start to suffer with gradient flow problems once the depth crosses 6 layers. To address this issue, they introduce residual connections among the LSTM layers in stack which greatly improves the gradient flow during backpropagation. They have used bi-directional encoder for the first layer since information required to translate certain words on the output side can appear anywhere on the source side. Model parallelism and data parallelism to speed up the training of their complex models and further details about their parallel implementation is provided in section 3.3.

In order to deal with the translation of rare words in the source, authors have used segmentation approaches to divide the words into sub-word units instead of just copying the rare words from source to target. They have provided the implementation details of this approach in section 4.1 and found that wordpieces achieve a balance between flexibility of characters and efficiency of words and the models achieve better BLEU scores when using wordpieces. For their mixed word/character model approach, they have used special prefixes to show the location of characters in a word and distinguish them from normal in-vocabulary characters. In section 5, they discuss the training criteria where they elaborate the shortcomings of using maximum-likelihood training and its weak robustness. They attempte to refine a model pretrained on the maximum likelihood objective to optimize the task reward which improves the results considerably. For their RL experiments, they have used GLEU scores instead of BLEU since it

correlates well with BLEU metric on corpus level but does not have any drawbacks for one sentence reward objective. In section 6, they discuss the quantized inference using reduced precision arithmetic technique which reduces the cost of inference of their models and provide efficiency improvements. Their main approach to speed up inference is with quantized arithmetic and hardware options available at Google. Also, they have added certain constraints to the models during training so that models are quantized with minimal impact on the output of the model. They have provided further training procedures for encoders and decoders in their models later in the section. Using CPU, GPU and TPU, they compare the inference speed and quality when decoding the WMT'14 English to French translation set. They provide further details about decoding using these units and since they found little difference in quality in models decoded on CPU and TPUs, they use CPUs to decode for model evaluation during training and experimentation and use TPUs to serve production traffic. For decodings, beam search is used to find the sequence Y that maximizes a score function s(Y,X) given a trained model and also introduce two refinements to the pure max-probability based beam search algorithm: a coverage penalty and length normalization. After the experiments with different models, they find that length normalization and coverage penalty are less effective models with RL refinement.

In section 8, they present their experimental results on WMT'14 English to French and English to German datasets. They also test GNMT on Google's translation production corpora, and compare the accuracy of model against human accuracy and best phrase-based machine translation production system for Google translate. Evaluation of models is done using standard BLEU score metric and side-by-side evaluations are carried out where they have human rates evaluate and compare the quality of two translations. For the training, they follow the classic data parallelism paradigm and apply gradient clipping since RNN models are built. They provide further training details, in the section and methods they use to address overfitting. Evaluation of models after maximum likelihood training is provided in section 8.4 where their best model WPK-32K achieves a BLEU score of 38.95. Since the models are optimized for log-likelihood of next step prediction which may not correlate well with the translation quality, they use RL refined models after normal maximum-likelihood training. Models are further ensembled and they carry out a four-way side-by-side human evaluation for comparison with other machine-translation systems. It is found that even though RL refinement achieves better BLEU scores, it barely improves the human impression of translation quality. Hence, for the production data, they do not use the RL-based model and also skip the dropout due to availability of large volume of training data. They further describe their experiments with human perception of translation quality and show that their model reduces the translation errors by more than 60% when compared to phrase-based systems.

Finally in the conclusion section they list their key findings and more importantly they show that their approach delivers high quality translations to much larger production data sets which have several order of magnitudes of large data.

Strengths

- The paper is well organized and the description of the architecture is detailed.
- Good amount of tables and figures are provided which is helpful in thoroughly understanding their approach and results.
- The methods used for evaluation are very robust and they have shown verified it in production level setting as well.

Weaknesses

- I think the paper is a bit more lengthier than necessary and some discussions which are done multiple times could be omitted.

Points of Confusion

- I am confused with section 6 which discusses about quantized inference and its mathematical workings.
- Why did they only start the residual connections from the third layer from bottom in the encoder and decoder instead of using it in all of the connections?

Discussion Questions

- What is the main difference between GLEU and BLEU scoring metrics and why GLEU doesn't have any drawbacks for the per sentence reward objective?
- One of the main benefits of using LSTMs was allowing proper gradient flow during backpropagation. Why is additional help like new residual connections needed for gradients if we are already using LSTMs in deeper networks?
- In order to allow maximum possible parallelization authors have used bi-directional connections in only the bottom encoder layer. Would it allow the networks to perform at the cost of less parallelization if bi-directional connections are used in other encoder layers as well?