Doctoral Thesis Review

Faculty of Mathematics and Physics, Charles University

Thesis author	Mgr. Jindřich Helcl
Thesis title	Non-Autoregressive Neural Machine Translation
Submission year	2021
Review author	Mgr. Martin Popel, Ph.D. **Role** Opponent
Department	Institute of Formal and Applied Linguistics

The submitted PhD thesis "Non-Autoregressive Neural Machine Translation" deals with a very topical issue of improving the decoding speed of NMT systems. While standard, autoregressive, NMT models generate the translation sequentially, conditioning each token on the previously generated tokens, non-autoregressive models generate all the tokens in one step, which can be parallelized and thus faster. A major drawback of non-autoregressive models is their lower translation quality. The goal of the thesis was to find a reasonable trade-off between decoding speed and translation quality.

The thesis consists of five chapters and a conclusion. The first chapter presents a brief introduction including a summary of MT and AI history. The second chapter provides a sound description of most popular NMT architectures (RNN and Transformer) and techniques. The author builds on strong knowledge gained by implementing most of the techniques himself within the Neural Monkey framework and teaching a seminar on the topic (Compendium of Neural Machine Translation).

The third chapter gives a survey of non-autoregressive NMT, its main principles, problems and a categorization of the approaches into four groups. The field is quite young (first publications are from 2018) and the overview covers even very recent publications (from 2021). I am not aware of any other overview of non-autoregressive NMT which would be so thorough and clearly written. I appreciate also the last section about evaluation methodology, which emphasizes that a fair comparison should present a Pareto frontier graphs (of decoding speed and translation quality) and include strong autoregressive baselines.

The fourth chapter describes the author's own contribution – the introduction of Connectionist Temporal Classification (CTC) loss into non-autoregressive NMT. CTC allows the model to produce more symbols than the number of tokens in the reference translation, but repeated symbols and blank tokens are ignored in the output, so a dynamic programming can be used during training for computing the sum of cross-entropy of all symbol sequences equivalent to the reference. This approach is quite different from the previous non-autoregressive NMT works and it influenced the field. The chapter is based on a paper published at EMNLP 2018 (Libovický and Helcl, 2018), which gained 67 citations within the three years, and on a preprint (Kasner, Libovický, Helcl, 2020), which describes a way to improve the translation fluency using an n-gram LM (autoregressive) rescoring at the cost of slightly slower decoding. Even with this improvement, the translation quality was substantially worse than the autoregressive baselines.

This motivated the experiments in the fifth chapter, where both decoding speed and translation quality was improved. Part of this success is due to faster implementation (C++ Marian) and stronger autoregressive baselines (with better data cleaning, model ensembling etc.), which were used as teacher models in knowledge distillation. While the preparation of teacher models described in Section 5.1 was a joint work with eight other authors (Chen et al., 2021), the distillation of non-autoregressive student models and integration of the CTC-loss into Marian was done by Jindřich Helcl (as aparent from GitHub log).

I appreciate the detailed and fair evaluation, which follows recent best practices (e.g. reporting COMET and chrF in addition to BLEU with SacreBLEU signatures and bootstrap confidence intervals). Table 5.9 shows that a speed-optimized autoregressive model (Edinburgh base) is Pareto-optimal in GPU and CPU batch translation (its translations are faster and of a higher quality than the translations by the CTC-based non-autoregressive models). In GPU online translation (i.e. batch size = 1), the non-autoregressive Large model is 2.4 times faster, but 7.5 BLEU worse than Edinburgh base. In Jindřich's own words: "*without future advancements in this field, optimized AR models will continue to achieve superior results over NAR models*" and "*many research studies concluding that NAR models are superior over AR models may be overclaiming*".

This thesis is written in comprehensible English and adequate academic style with a clear structure and only rare typos (most unfortunately in Equation 4.3).

Comments and questions

- I appreciate the plots of latency vs. source-sentence length (e.g. Figure 4.5). It would be useful to also report BLEU vs. source-sentence length (using a bigger test set and bucketing sentences by their length, or reporting some sentence-level metric instead BLEU).

  The author writes "*the optimal scenario for NAR models is a situation in which the system runs in online mode, i.e. with a batch size of 1*". and the final evaluation shows it is perhaps the *only* scenario where non-autoregressive models can be useful. However, in such scenarios (e.g. online subtitling or speech-to-speech MT), the latency of AR models for short sentences is sufficient; we care only about longer sentences. So we need to know what is the degradation in translation quality on such longer sentences.

- It would be worth discussing character-based NMT and document-level NMT, which both result in much longer sequences (than standard subword-based sentence-level NMT) and could thus benefit from non-autoregressive decoding. The question is whether non-autoregressive models could scale up to such long sequences without even bigger degradation in translation quality than reported in the thesis.

- It is not fully explained why are the optimized autoregressive models faster (and better) than the current non-autoregressive (or not much slower in online settings). The Conclusions chapter mentions quantization and pruning as a future work. I believe another reason is the asymmetry of Deep Encoder, Shallow Decoder (Kasai et al., 2020), while all non-autoregressive models in Chapter 5 have the same number layers in the encoder and decoder. A decoder with a single RNN (e.g. SRU) layer may be similarly fast as KenLM rescoring and faster than semi-regressive decoding.

- "*If the overall amount of padding used in a batch is large, the optimization step takes longer.*" Longer than what? The training time needed for each batch is not determined by the overall amount of padding used in a batch, but mostly by the length of the longest sentence (both source and target) and partly by the batch size because larger batches may overfill the GPU parallelization capabilities. The amount of padding is important for not "wasting" memory and for the overall training time (over all batches).

- The section about batching (on page 21) includes no references, but especially the second and fifth paragraph would deserve some.

- It is not clear whether the results of Gu et al. (2018) and Lee et al. (2018) in Table 4.3 are reimplementations (in Neural Monkey) or just reporting the scores provided by the original authors. In the former case, it would be nice to also report the decoding speed (latency and throughput using the same HW). In both cases, it should be reported if there were any differences in the training data and preprocessing. It seems suspicious that the issues with Romanian diacritics were encountered only in the CTC-based experiments. Section 4.5 mentions "*noisier dataset than what can be obtained*", but it is not obvious if there was any difference in the training data of Gu et al. (2018), Lee et al. (2018) and the CTC-based experiments.

- There are nine CTC-based NAR models in Table 4.3, but it is not clear which one is reported in Table 4.4. While weight averaging (done after training, but before inference) should have no impact on the speed and the effect of adding positional encoding should be negligible, I would expect the beam search to be slower and also enc-dec to be slower than the deep encoder.

- "*The models can still generate repeated words; for example, by interleaving them with the blank token.*" Why "for example"? What are the other ways of generating repeated subwords than interleaving them with one or more blank tokens? Section 4.1 says "*Optionally, identical outputs produced by multiple consecutive states may be merged*", but it is not clear if this option was used in the experiments or not. It is also not clear if the deduplication is done only on subword level or also on word level (cf. "*repeated words*").

- The effect of the splitting factor (set to 3 in all experiments) on speed and translation quality is not discussed.

- Table 4.7 shows latencies, but it is not clear for which translation direction (or an average of all four directions).

Conclusion

While the main results are mainly pessimistic regarding the future of non-autoregressive NMT models, I consider the thesis a valuable contribution to the field of efficient machine translation and I recommend it for defense. I believe the thesis and strong publication record clearly demonstrate the author's ability to conduct research independently and to present its results.

In Zubří, January 27th, 2022

Signature: