Language Modeling, Lexical Translation, Reordering: The Training Process of NMT through the Lens of Classical SMT

Elena Voita[1,2] **Rico Sennrich**[3,1] **Ivan Titov**[1,2]

[1]University of Edinburgh, Scotland [2]University of Amsterdam, Netherlands
[3]University of Zurich, Switzerland
lena-voita@hotmail.com sennrich@cl.uzh.ch ititov@inf.ed.ac.uk

Abstract

Differently from the traditional statistical MT that decomposes the translation task into distinct separately learned components, neural machine translation uses a single neural network to model the entire translation process. Despite neural machine translation being de-facto standard, it is still not clear how NMT models acquire different competences over the course of training, and how this mirrors the different models in traditional SMT. In this work, we look at the competences related to three core SMT components and find that during training, NMT first focuses on learning target-side language modeling, then improves translation quality approaching word-by-word translation, and finally learns more complicated reordering patterns. We show that this behavior holds for several models and language pairs. Additionally, we explain how such an understanding of the training process can be useful in practice and, as an example, show how it can be used to improve vanilla non-autoregressive neural machine translation by guiding teacher model selection.

1 Introduction

In the last couple of decades, the two main machine translation paradigms have been statistical and neural MT. Statistical MT (SMT) decomposes the translation task into several components (e.g., lexical translation probabilities, alignment probabilities, target-side language model, etc.) which are learned separately and then combined in a translation model. Differently, neural MT (NMT) models the entire translation process with a single neural network that is trained end-to-end.

Although joint training of all the components is one of the obvious NMT strengths, this is also one of its challenging aspects. While SMT models different competences with distinct model components and, therefore, can easily validate and/or improve each of them, NMT acquires these competences within the same network over the course of training. Even though previous work shows how to improve some of the competences in NMT, e.g., by using lexical translation probabilities, phrase memories, target-side LM, alignment information (Arthur et al., 2016; He et al., 2016; Tang et al., 2016; Wang et al., 2017; Zhang et al., 2017a; Dahlmann et al., 2017; Gülçehre et al., 2015; Gülçehre et al., 2017; He et al., 2016; Sriram et al., 2017; Dahlmann et al., 2017; Stahlberg et al., 2018; Mi et al., 2016b; Liu et al., 2016; Chen et al., 2016; Alkhouli et al., 2016; Alkhouli and Ney, 2017; Park and Tsvetkov, 2019; Song et al., 2020a among others), it is still not clear how and when NMT acquires these competences during training. For example, are there any stages where NMT focuses on different aspects of translation, e.g., fluency (agreement on the target side) or adequacy (i.e. connection to the source), or does it improve everything at the same rate? Does it learn word-by-word translation first and more complicated patterns later, or is there a different behavior? This is especially interesting in light of a recent work analyzing how NMT balances the two different types of context: the source and prefix of the target sentence (Voita et al., 2021). As it turns out, changes in NMT training are non-monotonic and form several distinct stages (e.g., stages changing direction from decreasing influence of source to increasing), which hints that the NMT training consists of stages with qualitatively different changes.

In this paper, we try to understand what happens in these stages by analyzing translations generated at different training steps. Specifically, we focus on the aspects related to the three core SMT components: target-side language modeling, lexical translation, and reordering. We find that during training, NMT focuses on these aspects in the specified above order. Intuitively, it starts by hallucinating frequent n-grams and sentences in the target language, then comes close to word-by-word

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8478–8491
November 7–11, 2021.

translation, and finally learns more complicated reordering patterns. We confirm these findings for several models, LSTM and Transformer, and different modeling paradigms, encoder-decoder and decoder-only, i.e. LM-style machine translation where a left-to-right language model is trained on the concatenation of source and target sentences.

Finally, we show how such an understanding of the training process can be useful in practice. Namely, we note that during a large part of training, a model's quality (e.g. BLEU and token-level predictive accuracy) changes little, but reordering becomes more complicated. This means that by using different training checkpoints, we can get high-quality translations of varying complexity, which is useful in settings where data complexity matters. For example, guiding teacher model selection for distillation in non-autoregressive machine translation (NAT) can improve the quality of a vanilla NAT model by more than 1 BLEU.

Our contributions are as follows:

- we show that during training, NMT undergoes the following three stages:
  - target-side language modeling;
  - learning how to use source and approaching word-by-word translation;
  - refining translations, visible by increasingly complex reorderings, but almost invisible to standard metrics (e.g. BLEU).
- we confirm our finding for different models and modeling paradigms;
- we explain how our analysis can be useful in practice and, as an example, show how it can improve a non-autoregressive NMT model.

2 Training Stages: The Two Viewpoints

In this section, we introduce two points of view on the NMT training process. The first one comes from previous work showing distinct stages in NMT training. These stages are formed by looking at a model's internal workings and changes in the way it balances source and target information when forming a prediction. The second point of view is from this work: we take model translations at different training steps and look at some of their aspects mirroring, in a way, core SMT components.

While these two points of view are complete opposites (one sees only the model's innermost workings, the other – only its output), only taken together they can fully describe the training process. We start from the first, abstract, stages, then show how these inner processes look on the outside and conclude with one of the immediate practical applications of our analysis (Section 6).

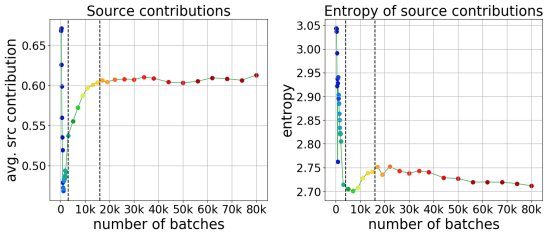


Figure 1: Contribution of source and entropy of source contributions. En-Ru. Vertical lines separate the stages.

2.1 The Abstract Viewpoint: Relative Token Contributions to NMT Predictions

The 'abstract' stages come from our previous work measuring how NMT balances the two different types of context: the source and prefix of the target sentence (Voita et al., 2021). We adapt one of the attribution methods, Layerwise Relevance Propagation (Bach et al., 2015), to the Transformer, and show how to evaluate the proportion of each token's influence for a given prediction. Then these relative token influences are used to evaluate the total contribution of the source (by summing up contributions of all source tokens) or to see whether the token contributions are more or less focused (by evaluating the entropy of these contributions).

Among other things, Voita et al. (2021) look at how the total source contribution and the entropy of source contributions change during training. We repeated these experiments for WMT14 En-Ru and En-De.[1] Figure 1 confirms previous observations: the training process is non-monotonic with several distinct stages, e.g. stages changing direction from decreasing influence of source to increasing.

These results suggest that during training, NMT undergoes stages of qualitatively different changes. For example, a decreasing and then increasing influence of the source likely indicates that the model first learns to rely on the target prefix more (i.e. to focus on target-side language modeling) and only after that focuses on the connection to the source (i.e. adequacy rather than fluency). While these hypotheses are reasonable, to confirm them we have to look not only at how model predictions are formed but also at the predictions themselves.

[1]Using the released code: https://github.com/lena-voita/the-story-of-heads.

2.2 The Practical Viewpoint: Model Translations

In this viewpoint, we are interested in changes in model output, i.e. translations. We measure:

- target-side language modeling scores;
- translation quality;
- monotonicity of alignments.

Note that these characteristics are related to three core components of the traditional SMT models: target-side language model, translation model, and reordering model. Although we are mainly interested in NMT models and, except for the language modeling scores, do not measure the quality of the corresponding SMT components directly, this relation to SMT is important. While machine translation is now mostly neural, it is still not clear how (e.g., in which order) those competences which used to be modelled with distinct components are now learned jointly within a single neural network.

3 Experimental Setting

3.1 Models, Data and Preprocessing

Models. We consider three models:

- Transformer encoder-decoder;
- LSTM encoder-decoder;
- Transformer decoder (LM-style NMT).

For the first model, we follow the setup of the Transformer base (Vaswani et al., 2017). LSTM encoder-decoder is a single-layer GNMT (Wu et al., 2016). The last model is the Transformer decoder trained as a left-to-right language model. In training, the model receives concatenated source and target sentences separated by a token-delimiter; in inference, it receives only the source sentence and the delimiter and is asked to continue generation.

Datasets. We use the WMT news translation shared task for English-German and English-Russian: for En-De, WMT 2014 with 5.8m sentence pairs, for En-Ru – 2.5m sentence pairs (parallel training data excluding UN and Paracrawl). Since our observations are similar for both languages, in the main text we show figures for one of them and in the appendix – for the other.

Preprocessing. The data is lowercased and encoded using BPE (Sennrich et al., 2016). We use separate source and target vocabularies of about 32k tokens for encoder-decoder models, and a joint vocabulary of about 50k tokens for LM-style models. For each experiment, we randomly choose 2/3 of the dataset for training and use the remaining 1/3 as a held-out set for analysis (see Section 3.3).

More details on hyperparameters, preprocessing, and training can be found in the appendix.

3.2 Target-Side LM Scores

For each of the models, we train 2-, 3-, 4- and 5-gram KenLM (Heafield, 2011)[2] language models on target sides of the corresponding training data (segmented with BPE). We report KenLM scores for the translations of the development sets.

3.3 Monotonicity of Alignments

To measure how the relative ordering of words in the source and its translation changes during training, we use two different scores used in previous work (Burlot and Yvon, 2018; Zhou et al., 2020). We evaluate the scores for two permutations of the source: the trivial monotonic alignment and the alignment inferred for the generated translation.

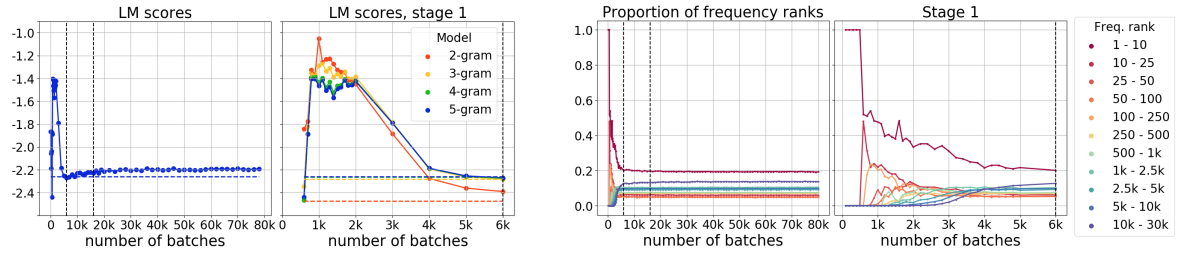
Fuzzy Reordering Score (Talbot et al., 2011) counts the number of chunks of contiguously aligned words and, intuitively, it is based on the number of times a reader would need to jump in order to read one reordering in the order proposed by the other. The score is between 0 and 1, where a larger score indicates more monotonic alignments.

Kendall tau distance (Kendall, 1938) is also called *bubble-sort distance* since it is equivalent to the number of swaps that the bubble sort algorithm would take to place one list in the same order as the other list. We evaluate the normalized distance: it is between 0 and 1, where 0 indicates the monotonic alignment.

The main difference between the scores is that the first one takes into account only the number of jumps, while the second also considers their distance. For a formal description of the scores and their differences, see the appendix.

Our setting. For each of the considered model checkpoints, we obtain datasets where the sources come from the held-out 1/3 of the original dataset, and targets are their translations. For these datasets, we infer alignments using `fast_align` (Dyer et al., 2013)[3].

[2] https://github.com/kpu/kenlm
[3] https://github.com/clab/fast_align



(a) (b)

Figure 2: (a) KenLM scores (horizontal dashed lines are the scores for the references); (b) proportion of tokens of different frequency ranks in model translations. En-Ru.

Source sentence: six months of construction works , that 's brutal .

Step	Model translation (German)	English (approximate translation from German)
100	, die ,	, the ,
300	, die , die , die , die , die , die , die , die , die , die , die ,	, the , the , the , the , the , the , the , the , the , the , the ,
700	die kommission , dass die kommission die kommission , die kommission , die kommission .	the commission , that the commission, the commission , the commission , the commission .
900	es ist , dass die kommission , dass die kommission .	it is that the commission that the commission .
1000	die europäische union ist , dass die eu .	the european union is that the eu .
1200	im rahmen der welt , die sich in der welt der welt .	within the framework of the world which is in the world of the world .
1700	zwei monaten , die sich in den letzten jahren ist .	two months, which is in recent years .
2000	sechs monate von fünf monaten , die die aus der ersten zeit .	six months of five months, which the from the first time .
3000	sechs monate des bauwerkes , das ist , dass der kampf .	six months of the building, that is that the struggle .
4000	sechs monate der bauarbeiten , die brutalen .	six months of construction, the brutal .
6000	sechs monate bauwerke , das brutale .	six months of buildings, the brutal .

Figure 3: Translations at different steps during training. En-De.

4 Transformer Training Stages

In this section, we discuss the standard encoder-decoder Transformer. In the next section, we mention differences with several other models.

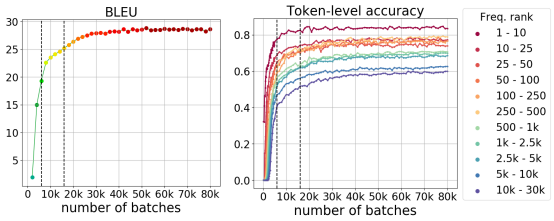
We first analyze the results for each of the three competences and then characterize the stages based on these practical observations. In all figures, we show the abstract stages with vertical lines to link the results to the changes in token contributions.

4.1 Target-Side Language Modeling

Figure 2a shows changes in the language modeling scores. We see that most of the change happens in the very beginning: the scores go up and peak much higher than that of the references. This means that the model generates sentences with very frequent n-grams rather than diverse texts similar to references. Indeed, Figure 2a (right) shows that for a part of the training (from 1k to 2k iterations), the scores for simpler models (e.g., 2-gram) are higher than for the more complicated ones (e.g., 5-gram). This means that generated translations tend to consist of frequent words and bigrams, but larger subsequences are not necessarily fluent.

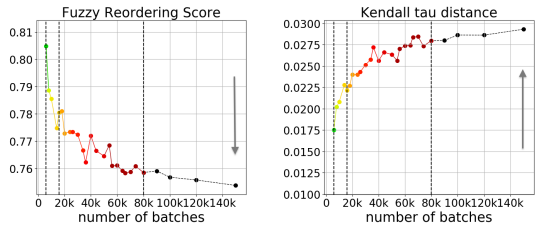
To illustrate this, we show how translations of one of the sentences evolve at the beginning of training (Figure 3). As expected, first the translations evolve from repetitions of frequent tokens to frequent bigrams and trigrams, and finally to longer frequent phrases. To make this more clear, we also show the proportion of tokens of different frequency ranks in generated translations (Figure 2b). First (iterations 0-500), all generated tokens are from the top-10 most frequent tokens, then only from the top-50, and only later less frequent tokens are starting to appear. From Figure 3 we see that this happens when the source comes into play: tokens related to the source become weaved into translations. Overall, this evolution from using short target-side contexts to longer ones and, subsequently, to using the source relates to works in computer vision discussing 'shortcut features' (Geirhos et al., 2020), as well as differences in the progression of extracting 'easy' and 'difficult' features during training (Hermann and Lampinen, 2020).

Note also that model translations converge to higher LM scores than references (Figure 2a). This is expected: compared to references, beam



(a) (b)

Figure 4: (a) BLEU score; (b) token-level accuracy (the proportion of cases where the correct next token is the most probable choice). WMT En-Ru.



(a) (b)

Figure 5: (a) fuzzy reordering score (for references: 0.6), (b) Kendall tau distance (for references: 0.06); WMT En-Ru. The arrows point in the direction of less monotonic alignments (more complicated reorderings).

search translations are simpler in various aspects, e.g. they are simpler syntactically, contain fewer rare tokens and less reordering (Burlot and Yvon, 2018; Ott et al., 2018; Zhou et al., 2020), and lead to more confident token contributions inside the model (Voita et al., 2021). For language models more generally, beam search texts are also less surprising than human ones (Holtzman et al., 2020).

To summarize, the beginning of training is mostly devoted to target-side language modeling: we see huge changes in the LM scores (Figure 2a), and the model hallucinates frequent n-grams (Figure 3). This agrees with the abstract stages shown in Figure 1: in the first stage, the total contribution of the source substantially decreases. This means that in the trade-off between information coming from the source and the target prefix, the model gives more and more priority to the prefix.

4.2 Translation Quality

Figure 4a shows the BLEU score on the development set during training. For a more fine-grained analysis, we also plot token-level predictive accuracy separately for target token frequency groups (Figure 4b). We see that both the BLEU score and accuracy become large very fast, e.g. after the first 20k iterations (25% of the training process), the scores are already good. What is interesting, is that the accuracy for frequent tokens reaches the maximum value (the score of the converged model) very quickly. This agrees with our previous observations in Figures 3 and 2b: at the beginning of training, the model generates frequent tokens more readily than the rare ones. Figure 4b further confirms this: the accuracy for the rare tokens improves slower than for the rest of them.

What is not clear, is what happens during the last half of the training (iterations from 40k to 80k): BLEU score improves only by 0.4, accuracy does not seem to change noticeably even for rare tokens, the proportion of generated tokens of different frequency ranks converges even earlier (Figure 2b), and patterns in token contributions also do not change much (Figure 1). This is what we are about to find out in the next section.

4.3 Monotonicity of Alignments

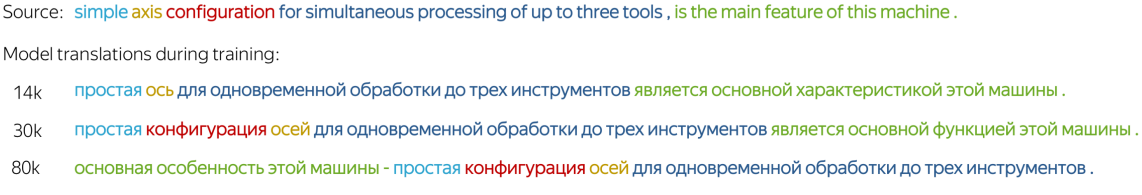
While it is known that, compared to references, beam search translations have more monotonic alignments (Burlot and Yvon, 2018; Zhou et al., 2020), it is not clear how monotonicity of alignments changes during model training. We show changes in the two reordering scores in Figure 5.[4]

We can say that during the second half of the training, the model is slowly refining translations, and, among the three competences we look at, the most visible changes are due to more complicated (i.e. less monotonic) reorderings. For example, as we already mentioned above, during this part of the training none of the scores we looked at so far changes much, whereas changes in both reordering scores are very substantial. The change in the fuzzy reordering score is only twice smaller than during the preceding stage. Moreover, the alignments keep changing and become less monotonic even after both BLEU and token-level accuracy (i.e. the metric that matches the model's training objective) converged, i.e. iterations after 80k (Figure 5).

Overall, we interpret this refinement stage as the model slowly learning to reduce interference from the source text (typical for human translation (Volansky et al., 2015) and exacerbated even more in NMT (Toral, 2019)): it learns to apply complex reorderings to more closely follow typical word order in the target language. This means that while language modeling improves more prominently during the first training stage, there is a long

[4]Note that we evaluate the scores starting not from the very beginning of training but after at least 6k updates. This is because evaluating monotonicity of alignments makes sense only when translations are reasonable.

(a) En-De



(b) En-Ru

Figure 6: Translations at different training steps. Same-colored chunks are approximately aligned to each other.

tail of less frequent and more nuanced patterns that the model learns later. Another example of such nuanced changes in translation not detected with standard metrics is context-aware NMT. Previous work has criticized using BLEU as a stopping criterion, showing that even when a model has converged in terms of BLEU, it continues to improve in terms of agreement with context (Voita et al., 2019b).

To illustrate changes during this last stage, we show two examples in Figure 6. On average, the translations at the beginning of the last stage tend to have the same word order as the corresponding source sentences: the alignments are highly monotonic. Formally, the similarity to the word-by-word translation is seen from the very low Kendall tau distance after 6k-14k training iterations (Figure 5b): this means that a very small number of permutations is needed to transform the trivial monotonic translation into the one produced by the model. Interestingly, at this point, some undertranslation errors can be explained via failures to perform a complex reordering. In the example in Figure 6b, the phrase 'axis configuration' cannot be translated into Russian preserving the same word order, which makes the model to omit the translation of 'configuration'.

4.4 Characterizing Training Stages

To summarize, the NMT training process can be described as undergoing the following three stages:

- target-side language modeling;
- learning how to use source and coming close to a word-by-word translation;
- refining translations, visible by an increase in complexity of the reorderings and almost invisible by standard evaluation (e.g. BLEU).

While the borders of these practical stages are not as strictly defined as the abstract ones with the changes of monotonicity in contribution graphs (Figure 1), these two points of view on the training process mirror each other very well. From the abstract point of view with token contributions, the model first starts to form its predictions based more on the prefix and ignores the source, then source influence increases quickly, then very little is going on (Figure 1). From the practical point of view with model translations, the model first hallucinates frequent tokens, then phrases, then sentences (mirrors source contributions going down), then quickly improves translation quality (mirrors source contribution going up), then little is going on according to the standard scores, but alignments become noticeably less monotonic. As we see, both points of view show the same kinds of processes from different perspective: from the inside and the outside of the model.

5 Other NMT Models

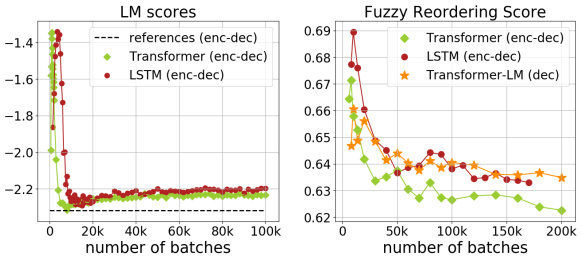
In this section, we compare different architectures within the same encoder-decoder framework (Transformer vs LSTM), and different frameworks with the Transformer architecture (encoder-decoder vs decoder-only). Overall, we find that all models follow the behavior described in Section 4.4; here we discuss some of their differences.

Transformer vs LSTM. As might be expected from the low BLEU scores (Table 1), LSTM translations are simpler than the Transformer ones. We see that they are less surprising according to the target-side language modeling scores (Figure 7a[5])

[5]Note that in Figure 7a, only the scores of the encoder-decoder models can be compared because of differences in

model	En-Ru	En-De
Transformer (enc-dec)	35.93	28.18
LSTM (enc-dec)	30.14	24.03
Transformer-LM (dec)	34.16	26.76

Table 1: BLEU scores: `newstest2014` for En-Ru and `newstest2017` for En-De.



(a) (b)

Figure 7: (a) target-side LM scores (5-gram), (b) fuzzy reordering score (for references: 0.5); WMT En-De.

and have more monotonic alignments (Figure 7b). Regarding the latter, it is not clear whether this is because of the lower model capacity or because LSTM has an inductive bias towards more monotonic alignments; we leave this to future work.

Encoder-decoder vs decoder-only. Table 1 shows that decoder-only (LM-style) NMT is not much worse than the standard encoder-decoder model, especially in the higher-resource setting (e.g., En-De). However, the decoder-only model has much simpler reordering patterns compared to the standard Transformer: its reordering scores are very close to the much weaker LSTM model (Figure 7b). One possible explanation is that the bidirectional nature of Transformer's encoder facilitates learning more complicated reorderings.

6 Practical Implications

We showed that during a large part of the training, the translation quality (e.g., BLEU) changes little, but the alignments become less monotonic. Intuitively, the translations become more complicated while their quality remains roughly the same.

One way to directly apply our analysis is to consider tasks and settings where data properties such as regularity and/or simplicity are important. For example, in neural machine translation, higher monotonicity of artificial sources was hypothesized to be a facilitating factor for back-translation (Burlot and Yvon, 2018); additionally, complexity of the distilled data is crucial for sequence-level distillation in non-autoregressive machine translation (Zhou et al., 2020). Such examples are not limited to machine translation: in emergent languages, languages with higher 'regularity' bring learning speed advantages for communicating neural agents (Ren et al., 2020).

In this section, we consider non-autoregressive NMT, and leave the rest to future work.

6.1 Non-Autoregressive Machine Translation

Non-autoregressive neural machine translation (NAT) (Gu et al., 2018) is different from the traditional NMT in the way it generates target sequences: instead of the standard approach where target tokens are predicted step-by-step by conditioning on the previous ones, NAT models predict the whole sequence simultaneously. This is possible only with an underlying assumption that the output tokens are independent from each other, which is unrealistic for natural language.

Fortunately, while this independence assumption is unrealistic for real references, it might be more plausible for simpler sequences, e.g. artificially generated translations. That is why targets for NAT models are usually not references but beam search translations of the standard autoregressive NMT (which, as we already mentioned above, are simpler than references in many aspects). This is called *sequence-level knowledge distillation* (Kim and Rush, 2016), and it is currently one of the de-facto standard parts of the NAT training pipelines (Gu et al. (2018); Lee et al. (2018); Ghazvininejad et al. (2019) to name a few).

Recently Zhou et al. (2020) showed that the quality of a NAT model strongly depends on the complexity of the distilled data, and changing this complexity can improve the model. Since distilled data consists of translations from a standard autoregressive teacher, our analysis gives a very simple way of modifying the complexity of this data. While usually a teacher is a fully converged model, we propose to use as teachers intermediate checkpoints during training. Since during a large part of training, NMT quality (e.g., BLEU) changes little, but the alignments become less monotonic, earlier checkpoints can produce simpler and more monotonic translations. We hypothesize that these translations are more suitable as targets for NAT models, and we confirm this with the experiments.

model vocabulary (see Section 3). In the appendix, we show scores for all three models.

6.2 Setting

Following previous work (Zhou et al., 2020), we train the same NAT model on their preprocessed dataset[6] and vary only distilled targets.

Model. The model is the re-implemented by Zhou et al. (2020) version of the vanilla NAT by Gu et al. (2018). For more details, see appendix.

Dataset. The dataset is WMT14 English-German (En-De) with newstest2013 as the validation set and newstest2014 as the test set, and BPE vocabulary of 37,000. We use the preprocessed dataset and the vocabularies released by Zhou et al. (2020).

Distilled targets. The teacher is the standard Transformer-base from `fairseq` (Ott et al., 2019). For the baseline distilled dataset, we use the fully converged model (in this case, the model after 200k updates). For other datasets, we use earlier checkpoints.

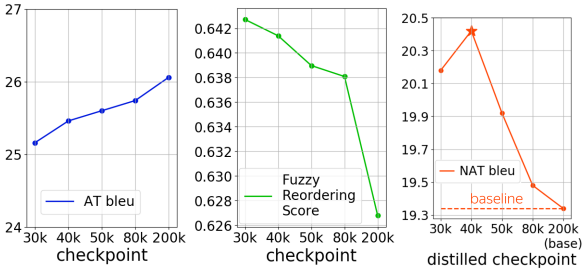
Evaluation. We average the last 10 checkpoints.

6.3 Experiments

Figure 8c shows the BLEU scores for NAT models trained with distilled data obtained from different teacher's checkpoints; the baseline is the fully converged model (200k iterations). We see that by taking an earlier checkpoint, after 40k iterations, we improve NAT quality by 1.1 BLEU. For this checkpoint, the teacher's BLEU score is not much lower than that of the final model (Figure 8a), but the reorderings are much simpler (a higher fuzzy reordering score in Figure 8b).

To vary the complexity of the distilled data, Zhou et al. (2020) proposed to apply either Born-Again networks (BANs) (Furlanello et al., 2018) or mixture-of-experts (MoE) (Shen et al., 2019). Unfortunately, MoE is rather complicated and requires careful hyperparameter tuning (Shen et al., 2019), and BANs are time- and resource-consuming. They involve training the AT model till convergence and then translating the training data to get a distilled dataset; this happens in several iterations (e.g., 5-7) using for training the latest generated dataset. Compared to these methods, our approach is extremely simple and does not require a lot of computational resources (e.g., instead of fully training the AT teacher several times as in BANs, our approach requires only to partially train one AT teacher).

[6]We used the code and the data from https://github.com/pytorch/fairseq/tree/master/examples/nonautoregressive_translation.



(a) (b) (c)

Figure 8: (a) BLEU score of the AT Transformer-base (teacher for distillation); (b) fuzzy reordering score for the distilled training data obtained from checkpoints of the AT teacher; (c) BLEU scores for the vanilla NAT model trained on different distilled data.

Note that in this work, we provide these experiments mainly to illustrate how our analysis can be useful in the settings where data complexity matters and, therefore, limit ourselves to only using different teacher checkpoints. Future work, however, can investigate possible combinations with other approaches. For example, to further improve quality, our method can be combined with the Born-Again networks while still requiring fewer resources due to only partial training of the teachers.

7 Additional Related Work

Other work connecting neural and traditional approaches include modeling modifications, such as modeling coverage and/or fertility (Tu et al., 2016; Mi et al., 2016a; Cohn et al., 2016; Feng et al., 2016) and several other modifications (Zhang et al., 2017b; Stahlberg et al., 2017; Huang et al., 2018), analysis of the relation between attention and word alignments (Ghader and Monz, 2017), and word alignment induction from NMT models (Li et al., 2019; Garg et al., 2019; Song et al., 2020b; Zenkel et al., 2020; Chen et al., 2020).

Previous analysis of NMT learning dynamics include analyzing how the trainable parameters affect an NMT model (Zhu et al., 2020) and looking at the speed of learning specific discourse phenomena in context-aware NMT (Voita et al., 2019b,a).

8 Conclusions

We analyze how NMT acquires different competencies during training and look at the competencies related to three core SMT components. We find that NMT first focuses on learning target-side language modeling, then improves translation quality

approaching word-by-word translation, and finally learns more complicated reordering patterns. We show that such an understanding of the training process can be useful in settings where data complexity matters and illustrate this for non-autoregressive MT; other tasks can be considered in future work. Additionally, our results can contribute to the discussion of (i) 'easy' and 'difficult' task-relevant features, including 'shortcut features', and (ii) the limitations of the BLEU score.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. Lena is supported by the Facebook PhD Fellowship. Rico Sennrich acknowledges support of the Swiss National Science Foundation (MUTAMUR; no. 176727). Ivan Titov acknowledges support of the European Research Council (ERC StG BroadSem 678254), Dutch National Science Foundation (VIDI 639.022.518) and EU Horizon 2020 (GoURMET, no. 825299).

References

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas. Association for Computational Linguistics.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. 2017. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1420, Copenhagen, Denmark. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Shi Feng, Shujie Liu, Nan Yang, Mu Li, Ming Zhou, and Kenny Q. Zhu. 2016. Improving attention modeling with implicit distortion and fertility for machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3082–3092, Osaka, Japan. The COLING 2016 Organizing Committee.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Comput. Speech Lang.*, 45:137–148.

W. He, Zhongjun He, Hua Wu, and H. Wang. 2016. Improved neural machine translation with smt features. In *AAAI*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Katherine Hermann and Andrew Lampinen. 2020. What shapes feature representations? exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, pages 9995–10006. Curran Associates, Inc.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2018. Towards neural phrase-based machine translation. In *International Conference on Learning Representations*.

M.G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representation (ICLR 2015)*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016a. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016b. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965, Stockholmsmässan, Stockholm Sweden. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*

(Demonstrations), pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Chan Young Park and Yulia Tsvetkov. 2019. Learning to generate word- and phrase-embeddings for efficient phrase-based neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 241–248, Hong Kong. Association for Computational Linguistics.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. 2020. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5719–5728. PMLR.

Kai Song, K. Wang, H. Yu, Y. Zhang, Zhongqiang Huang, Wei-Hua Luo, Xiangyu Duan, and M. Zhang. 2020a. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *AAAI*.

Kai Song, Xiaoqing Zhou, Heng Yu, Zhongqiang Huang, Yue Zhang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020b. Towards better word alignment in transformer. volume 28, pages 1801–1812.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 204–211, Brussels, Belgium. Association for Computational Linguistics.

Felix Stahlberg, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2017. Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 362–368, Valencia, Spain. Association for Computational Linguistics.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz Och. 2011. A lightweight evaluation framework for machine translation reordering. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 12–21, Edinburgh, Scotland. Association for Computational Linguistics.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip L. H. Yu. 2016. Neural machine translation with external phrase memory.

Antonio Toral. 2019. Post-editese: an exacerbated translationese. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 273–281, Dublin, Ireland. European Association for Machine Translation.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Los Angeles.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Vered Volansky, Noam Ordan, and S. Wintner. 2015. On the features of translationese. *Digit. Scholarsh. Humanit.*, 30:98–118.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Copenhagen, Denmark. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017a. Prior knowledge integration for neural machine translation using posterior regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1514–1523, Vancouver, Canada. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2017b. Improving neural machine translation through phrase-based forced decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 152–162, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

Conghui Zhu, Guanlin Li, Lemao Liu, Tiejun Zhao, and Shuming Shi. 2020. Understanding learning dynamics for neural machine translation.

A Experimental Setting

A.1 Data preprocessing

Translation pairs were batched together by approximate sequence length. Each training batch contained a set of translation pairs containing approximately 32000 source tokens.[7]

A.2 Model parameters

Transformer (encoder-decoder). We follow the setup of Transformer base model (Vaswani et al., 2017). More precisely, the number of layers in the encoder and in the decoder is $N = 6$. We employ $h = 8$ parallel attention layers, or heads. The dimensionality of input and output is $d_{model} = 512$, and the inner-layer of a feed-forward networks has dimensionality $d_{ff} = 2048$. We use regularization as described in (Vaswani et al., 2017).

Transformer (decoder). The difference from the previous model is that the decoder has 12 layers.

LSTM (encoder-decoder) is a single-layer GNMT (Wu et al., 2016) with the input and output dimensionality of 512 and hidden sizes of 1024.

A.3 Optimizer

The optimizer we use is the same as in (Vaswani et al., 2017). We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\varepsilon = 10^{-9}$. We vary the learning rate over the course of training, according to the formula:

$$l_{rate} = scale \cdot \min(step\_num^{-0.5},$$

$$step\_num \cdot warmup\_steps^{-1.5})$$

We use $warmup\_steps = 16000$, $scale = 4$.

B Monotonicity of Alignments

To measure how the relative ordering of words in the source and target sentences changes during training, we use two different scores: fuzzy reordering score (Talbot et al., 2011) and Kendall tau distance. We evaluate both scores for two permutations of the source sentence σ_1 and σ_2, where $\sigma_1$is the trivial monotonic alignment and σ_2 – the alignment inferred for the generated translation.

[7]This can be reached by using several of GPUs or by accumulating the gradients for several batches and then making an update.

Fuzzy Reordering Score aligns each word in σ_1 to an instance of itself in σ_2 taking the first unmatched instance of the word if there is more than one. If C is the number of chunks of contiguously aligned words and M is the number of words in the source sentence, then the fuzzy reordering score is computed as

$$FRS(\sigma_1, \sigma_2) = 1 - \frac{C-1}{M-1}. \quad (1)$$

This metric assigns a score between 0 and 1, where 1 indicates that the two reorderings are identical. Intuitively, C is the number of times a reader would need to jump in order to read the reordering σ_1 in the order proposed by σ_2. A larger fuzzy reordering score indicates more monotonic alignments.

Kendall tau distance counts the number of pairwise disagreements between two ranking lists. The larger the distance, the more dissimilar the two lists are. Kendall tau distance is also called *bubble-sort distance* since it is equivalent to the number of swaps that the bubble sort algorithm would take to place one list in the same order as the other list. We evaluate the normalized distance, i.e. for a list of length n it is normalized by $\frac{n(n-1)}{2}$. The normalized score is between 0 and 1, where 0 indicates that the two reorderings are identical.

Differences between the scores. While the first score counts only the number of chunks of contiguously aligned words, the second one takes into account only how distant the changes are. For example, let us consider two reorderings: $(2, 1, 4, 3, 6, 5)$ and $(4, 5, 6, 1, 2, 3)$. While for the fuzzy reordering score the least monotonic reordering is the first (more jumps for a reader), for the Kendall tau score – the second (requires more permutations to reorder). As we will see in Section 4.3, results for the two scores are similar.

Our setting. We take sentences of at least 2 words for the fuzzy reordering score and at least 10 tokens for the Kendall tau distance.

C Transformer Training Stages

Figure 9 shows the abstract stages for En-De, Figures 10-13 provide the results from Section 4 for the other language pair (En-De).

D Other Models

Figure 14 is a version of the Figure 7a from the main text, but with the scores for all three mod-

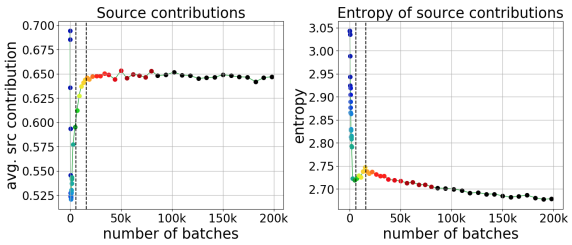


Figure 9: Left: contribution of source, right: entropy of source contributions. En-De. Vertical lines separate the stages.

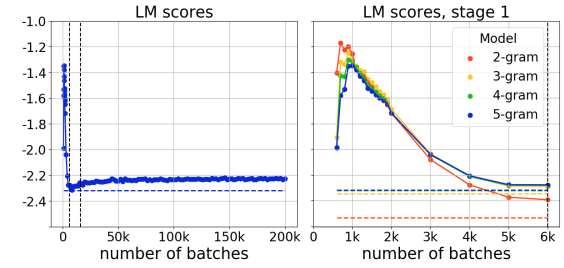


Figure 10: KenLM scores. Left: 5-gram model, all training stages; right: different models, the first stage. Horizontal lines show the scores for the references. En-De.

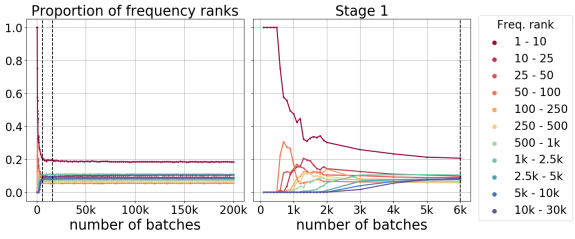
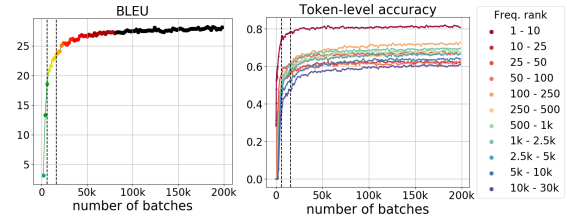


Figure 11: Proportion of tokens of different frequency ranks in model translations. En-De.



(a) (b)

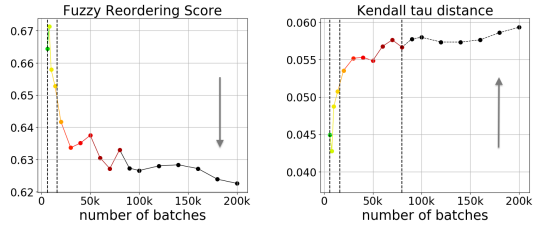
Figure 12: (a) BLEU score; (b) token-level accuracy (the proportion of cases where the correct next token is the most probable choice). WMT En-De.

els. Figure 15 provides corresponding results for the other language pair (En-Ru). Note that in Figure 15b the reordering score for the LSTM model stops earlier: this is because the LSTM model converges earlier than other models.

E Practical Applications

E.1 Experimental Setting

Model. The model is the re-implemented by Zhou et al. (2020) version of the vanilla NAT



(a) (b)

Figure 13: (a) fuzzy reordering score (for references: 0.5), (b) Kendall tau distance (for references: 0.08); WMT En-De. The arrows point in the direction of less monotonic alignments (more complicated reorderings).

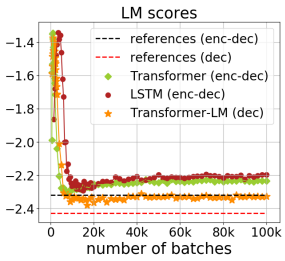
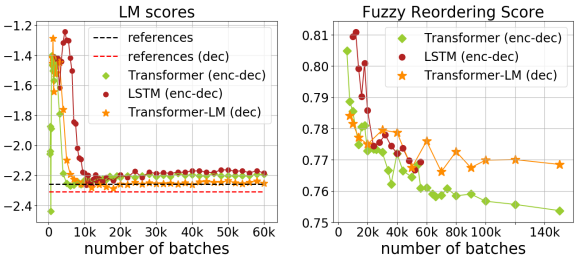


Figure 14: Target-side LM scores (5-gram); En-De.



(a) (b)

Figure 15: (a) target-side LM scores, (b) fuzzy reordering score (for references: 0.6); WMT En-Ru.

by Gu et al. (2018). Namely, instead of modeling fertility as described in the original paper, Zhou et al. (2020) monotonically copy the encoder embeddings to the input of the decoder. We used the code released by Zhou et al. (2020).[8]

Training. For all experiments, we follow the setting by Zhou et al. (2020). Note that in their work, training NAT models required 32 GPUs. In our setting, we ensured the same batch size by accumulating gradients for several batches (in `fairseq`, this is done using the `-update-freq` option).

NAT Inference. Following previous work, for this vanilla NAT model we use a straightforward decoding algorithm which simply picks the `argmax` at every position.

[8] https://github.com/pytorch/fairseq/tree/master/examples/nonautoregressive_translation