Analyzing the Source and Target Contributions to Predictions in Neural Machine Translation

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

In Neural Machine Translation (and, more generally, conditional language modeling), the generation of a target token is influenced by two types of context: the source and the prefix of the target sequence. While many attempts to understand the internal workings of NMT models have been made, none of them explicitly evaluates relative source and target contributions to a generation decision. We argue that this relative contribution can be evaluated by adopting a variant of Layerwise Relevance Propagation (LRP). Its underlying 'conservation principle' makes relevance propagation unique: differently from other methods, it evaluates not an abstract quantity reflecting token importance, but the proportion of each token's influence. We extend LRP to the Transformer and conduct an analysis of NMT models which explicitly evaluates the source and target relative contributions to the generation process. We analyze changes in these contributions when conditioning on different types of prefixes, when varying the training objective or the amount of training data, and during the training process. We find that models trained with more data tend to rely on source information more and to have more sharp token contributions; the training process is non-monotonic with several stages of different nature.[1]

1 Introduction

With the success of neural approaches to natural language processing, analysis of NLP models has become an important and active topic of research. In NMT, approaches to analysis include probing for linguistic structure (Belinkov et al., 2017; Conneau et al., 2018), evaluating via contrastive translation pairs (Sennrich, 2017; Burlot and Yvon, 2017; Rios Gonzales et al., 2017; Tang et al., 2018), inspecting model components, such as attention (Ghader and Monz, 2017; Voita et al., 2018; Tang et al., 2018; Raganato and Tiedemann, 2018; Voita et al., 2019) or neurons (Dalvi et al., 2019; Bau et al., 2019), among others.

Unfortunately, although a lot of work on model analysis has been done, a question of how the NMT predictions are formed remains largely open. Namely, the generation of a target token is defined by two types of context, source and target, but there is no method which explicitly evaluates the relative contribution of source and target to a given prediction. The ability to measure this relative contribution is important for model understanding since previous work showed that NMT models often fail to effectively control information flow from source and target contexts. For example, adding context gates to dynamically control the influence of source and target leads to improvement for both RNN (Tu et al., 2017; Wang et al., 2018) and Transfomer (Li et al., 2020) models. A more popular example is a model's tendency to generate hallucinations (fluent but inadequate translations); it is usually attributed to the inappropriately strong influence of target context. Several works observed that, when hallucinating, a model fails to properly use source: it produces a deficient attention matrix, where almost all the probability mass is concentrated on uninformative source tokens (EOS and punctuation) (Lee et al., 2018; Berard et al., 2019).

We argue that a natural way to estimate how the source and target contexts contribute to generation is to apply Layerwise Relevance Propagation (LRP) (Bach et al., 2015) to NMT models. LRP redistributes the information used for a prediction between all input elements keeping the total contribution constant. This 'conservation principle' makes relevance propagation unique: differently from other methods estimating influence of individual tokens (Alvarez-Melis and Jaakkola, 2017; He

[1]We release the code at https://github.com/lena-voita/the-story-of-heads.

et al., 2019a; Ma et al., 2018), LRP evaluates not an abstract quantity reflecting a token importance, but the proportion of each token's influence.

We extend one of the LRP variants to the Transformer and conduct the first analysis of NMT models which explicitly evaluates the source and target relative contributions to the generation process. We analyze changes in these contributions when conditioning on different types of prefixes (reference, generated by a model or random translations), when varying training objective or the amount of training data, and during the training process. We show that models suffering from exposure bias are more prone to over-relying on target history (and hence to hallucinating) than the ones where the exposure bias is mitigated. When comparing models trained with different amount of data, we find that extra training data teaches a model to rely on source information more heavily and to be more confident in the choice of important tokens. When analyzing the training process, we find that changes in training are non-monotonic and form several distinct stages (e.g., stages changing direction from decreasing influence of source to increasing).

Our key contributions are as follows:

- we show how to use LRP to evaluate the relative contribution of source and target to NMT predictions;
- we analyze how the contribution of source and target changes when conditioning on different types of prefixes: reference, generated by a model or random translations;
- we show that models suffering from exposure bias are more prone to over-relying on target history (and hence to hallucinating);
- we find that (i) with more data, models rely on source information more and have more sharp token contributions, (ii) the training process is non-monotonic with several distinct stages.

2 Layer-wise Relevance Propagation

Layer-wise relevance propagation is a framework which decomposes the prediction of a deep neural network computed over an instance, e.g. an image or sentence, into relevance scores for single input dimensions of the sample such as subpixels of an image or neurons of input token embeddings. The original LRP version was developed for computer vision models (Bach et al., 2015) and is not directly applicable to the Transformer (e.g., to the attention layers). In this section, we explain the general idea behind LRP, specify which of the existing LRP variants we use, and show how to extend LRP to the NMT Transformer model.[2]

2.1 General Idea: Conservation Principle

In its general form, LRP assumes that the model can be decomposed into several layers of computation. The first layer are the inputs (for example, the pixels of an image or tokens of a sentence), the last layer is the real-valued prediction output of the model f. The l-th layer is modeled as a vector $x^{(l)} = (x_i^{(l)})_{i=1}^{V(l)}$ with dimensionality $V(l)$. Layer-wise relevance propagation assumes that we have a relevance score $R_i^{(l+1)}$ for each dimension $x_i^{(l+1)}$ of the vector x at layer $l+1$. The idea is to find a relevance score $R_i^{(l)}$ for each dimension $x_i^{(l)}$ of the previous layer l such that the following holds:

$$f = \ldots = \sum_i R_i^{(l+1)} = \sum_i R_i^{(l)} = \ldots = \sum_i R_i^{(1)}. \quad (1)$$

This equation represents a *conservation principle*, which LRP exploits to back-propagate the prediction. Intuitively, this means that the total contribution of neurons at each layer is constant.

2.2 Redistribution Rules

Assume that we know the relevance $R_j^{(l+1)}$ of a neuron j at network layer $l+1$ for the prediction $f(x)$. Then we would like to decompose this relevance into messages $R_{i\leftarrow j}^{(l,l+1)}$ sent from the neuron j at layer $l+1$ to each of its input neurons i at layer l. For the conservation principle to hold, these messages $R_{i\leftarrow j}^{(l,l+1)}$ have to satisfy the constraint:

$$R_j^{(l+1)} = \sum_i R_{i\leftarrow j}^{(l,l+1)}. \quad (2)$$

Then we can define the relevance of a neuron i at layer l by summing all messages from neurons at layer $(l+1)$:

$$R_i^{(l)} = \sum_j R_{i\leftarrow j}^{(l,l+1)}. \quad (3)$$

Equations (2) and (3) define the propagation of relevance from layer $l+1$ to layer l. The only thing that is missing is specific formulas for computing the messages $R_{i\leftarrow j}^{(l,l+1)}$. Usually, the message $R_{i\leftarrow j}^{(l,l+1)}$ has the following structure:

$$R_{i\leftarrow j}^{(l,l+1)} = v_{ij} R_j^{(l+1)}, \quad \sum_i v_{ij} = 1. \quad (4)$$

[2] Previous work applying one of the LRP variants to NMT (Ding et al., 2017; Voita et al., 2019) do not describe extensions beyond the original LRP rules (Bach et al., 2015).

Several versions of LRP satisfying equation (4) (and, therefore, the conservation principle) have been introduced: LRP-ε, LRP-$\alpha\beta$ and LRP-γ (Bach et al., 2015; Binder et al., 2016; Montavon et al., 2019). We use LRP-$\alpha\beta$ (Bach et al., 2015; Binder et al., 2016), which defines relevances at each step in such a way that they are positive.

Rule for relevance propagation: the $\alpha\beta$-rule. Let us consider the simplest case of linear layers with non-linear activation functions, namely

$$z_{ij} = x_i^{(l)} w_{ij}, \quad z_j = \sum_i z_{ij} + b_i, \quad x_j^{(l+1)} = g(z_j),$$

where w_{ij} is a weight connecting the neuron $x_i^{(l)}$ to neuron $x_j^{(l+1)}$, b_j is a bias term, and g is a non-linear activation function. Let

$$z_j^+ = \sum_i z_{ij}^+ + b_j^+, \quad z_j^- = \sum_i z_{ij}^- + b_j^-,$$

where $\square^+ = \max(0, \square)$ and $\square^- = \min(0, \square)$. Then the $\alpha\beta$-rule (Bach et al., 2015; Binder et al., 2016) is given by the equation

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \cdot \left(\alpha \cdot \frac{z_{ij}^+}{z_j^+} + \beta \cdot \frac{z_{ij}^-}{z_j^-} \right), \quad (5)$$

where $\alpha + \beta = 1$. Note that all terms in the brackets are always positive: negative signs of z_j^- and z_{ij}^- cancel out when evaluating the ratio.

This propagation method allows to control manually the importance of positive and negative evidence by choosing different α and β. For example, $\alpha, \beta = \frac{1}{2}$ treats positive and negative contributions as equally important, while $\alpha = 1$, $\beta = 0$ considers only positive contributions. In our experiments, we use $\alpha = 1$, $\beta = 0$.

Note that (5) is directly applicable to all layers for which there exist functions g_j and h_{ij} such that

$$x_j^{(l+1)} = g_j \left(\sum_i h_{ij}(x_i^{(l)}) \right). \quad (6)$$

These layers include linear, convolutional and max-pooling operations. Additionally, pointwise monotonic activation functions g_j (e.g., ReLU) are ignored by LRP (Bach et al., 2015).

Propagating relevance through attention layers. For the structures that do not fit the form (6) (e.g., softmax operation, layer normalization (Ba et al., 2016), residual connections), the weighting v_{ij} can be obtained by performing a first order Taylor expansion of a neuron $x_j^{(l+1)}$ (Bach et al., 2015; Binder et al., 2016).

For attention layers in the Transformer, we extend the approach by Binder et al. (2016). Namely, let $x_j^{(l+1)} = f(x^{(l)})$, $f(x) = f(x_1, \ldots, x_n)$. Then by Taylor expansion at some point $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$, we get

$$f(\hat{x}) \approx f(x^{(l)}) + \sum_{i\leftarrow j} \frac{\partial f}{\partial x_i}(x^{(l)}) \cdot (\hat{x}_i - x_i^{(l)}),$$

$$x_j^{(l+1)} = f(x^{(l)}) \approx f(\hat{x}) + \sum_{i\leftarrow j} \frac{\partial f}{\partial x_i}(x^{(l)}) \cdot (x_i^{(l)} - \hat{x}_i).$$

Elements of the sum can be assigned to incoming neurons, and the zero-order term can be redistributed equally between them. This leads to the following decomposition:

$$z_{ij} = \frac{1}{n} f(\hat{x}) + \frac{\partial f}{\partial x_i}(x^{(l)}) \cdot (x_i^{(l)} - \hat{x}_i). \quad (7)$$

We use the zero vector in place of $\hat{x}$. Equation (7), along with the standard redistribution rules (5), defines relevance propagation for complex non-linear layers. In the Transformer, we apply equation (7) to the softmax operations in the attention layers; all other operations inside the attention layers are linear functions, and the rule (5) can be used. We also apply equation (7) to layer normalization operations and residual connections.

2.3 LRP for Conditional Language Models

Given a source sequence $x = (x_1, \ldots, x_S)$ and a target sequence $y = (y_1, \ldots, y_T)$, standard autoregressive NMT models (or, in a more broad sense, conditional language models) are trained to predict words in the target sequence, word by word. Formally, at each generation step such models predict $p(y_t | x_{1:S}, y_{1:t-1})$ relying on both source tokens $x_{1:S}$ and already generated target tokens $y_{1:t-1}$. Using LRP, we evaluate relative contribution of all tokens, source and target, to the current prediction.

Propagating through decoder and encoder. At first glance, it can be unclear how to apply a layer-wise method to a not completely layered architecture (such as encoder-decoder). This, however, is rather straightforward and is done in two steps:

1. total relevance is propagated through the decoder. Since the decoder uses representations from the final encoder layer, part of the relevance 'leaks' to the encoder; this happens at each decoder layer;

2. relevance leaked to the encoder is propagated through the encoder layers.

The total contribution of neurons in each decoder layer is not preserved (part of the relevance leaks to the encoder), but the total contribution of all tokens – across the source and the target prefix – remains equal to the model prediction.

We evaluate relevance of input neurons to the top-1 logit predicted by a model. Then token relevance (or its contribution) is the sum of relevances of its neurons.

Notation. Without loss of generality, we can assume that the total relevance for each prediction equals 1.[3] Let us denote by $\mathrm{R}_t(x_i)$ and $\mathrm{R}_t(y_j)$ the contribution of source token x_i and target token y_j to the prediction at generation step t, respectively. Then source and target contributions are defined as

$$\mathrm{R}_t(\text{source}) = \sum_i \mathrm{R}_t(x_i), \quad \mathrm{R}_t(\text{target}) = \sum_{j=1}^{t-1} \mathrm{R}_t(y_j).$$

Note that $\forall\, t \;\; \mathrm{R}_t(\text{source}) + \mathrm{R}_t(\text{target}) = 1$; $\mathrm{R}_1(\text{source}) = 1$, $\mathrm{R}_1(\text{target}) = 0$, and $\forall j \geq t \;\; \mathrm{R}_t(y_j) = 0$.

3 Experimental setting

Model. We follow the setup of Transformer base model (Vaswani et al., 2017) with the standard training setting. More details on hyperparameters and the optimizer can be found in the appendix.

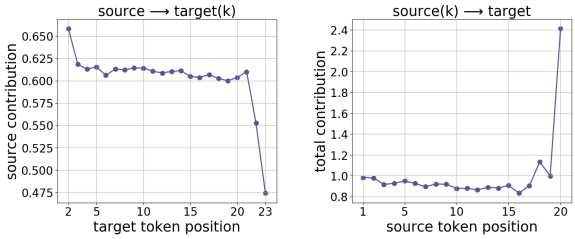
Data. We use random subsets of the WMT14 En-Fr dataset of different size: 1m, 2.5m, 5m, 10m, 20m, 30m sentence pairs. In Sections 4 and 7, we report results for the model trained on the 1m subset. In Section 6, we show how the results depend on the amount of training data.

Evaluating LRP. The $\alpha\beta$-LRP we use requires choosing values for α and β, $\alpha + \beta = 1$. We consider only positive contributions, i.e. we choose $\alpha = 1, \beta = 0$.[4]

Reporting results. All presented results are averaged over an evaluation dataset of 1000 sentence pairs. In each evaluation dataset, all examples have the same number of tokens in the source, as well as in the target (e.g., 20 source and 23 target tokens; the exact number for each experiment is clear from the results).[5]

[3] More formally, if we evaluate relevance for top-1 logit predicted by a model, then the total relevance is equal to the value of this logit. However, the conservation principle allows us to assume that this logit is equal to 1 and to consider relative contributions.

[4] In preliminary experiments, we performed visual inspection of the contribution heatmaps and observed that $\alpha, \beta = \frac{1}{2}$ (i.e., using also negative contributions) do not lead to reasonable contribution patterns.



(a) (b)

Figure 1: (a) contribution of the whole source at each generation step; (b) total contribution of source tokens at each position to the whole target sentence.

4 Getting Acquainted

In this section, we explain general patterns in model behavior and illustrate the usage of LRP by evaluating different statistics within a single model. Later, we will show how these results change when varying the amount of training data (Section 6) and during model training (Section 7).

4.1 Changes in contributions

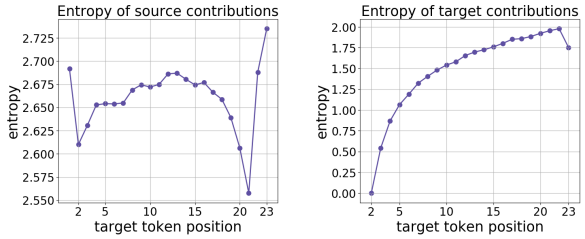
Here we evaluate changes in the source contribution during generation, and in contributions of source tokens at different positions to entire output.

Source ⟶ target(k). For each generation step t, we evaluate total contribution of source $\mathrm{R}_t(\text{source})$. Note that this is equivalent to evaluating total contribution of prefix since $\mathrm{R}_t(\text{prefix}) = 1 - \mathrm{R}_t(\text{source})$ (Section 2.3).

Results are shown in Figure 1(a).[6] We see that, during the generation process, the influence of source decreases (or, equivalently, the influence of the prefix increases). This is expected: with a longer prefix, the model has less uncertainty in deciding which source tokens to use, but needs to control more for fluency. There is also a large drop of source influence for the last token: apparently, to generate the EOS token, the model relies on prefix much more than when generating other tokens.

[5] Note that we have to fix the number of tokens in the source and target to get reliable comparisons. We choose sentences of length 20 and 23 because these are among the most frequent sentence lengths in the dataset. We also looked at sentences with 16, 25, 29 tokens – observed patterns were the same.

[6] Since the first token is always generated solely relying on the source, we plot starting from the second token.



(a) (b)

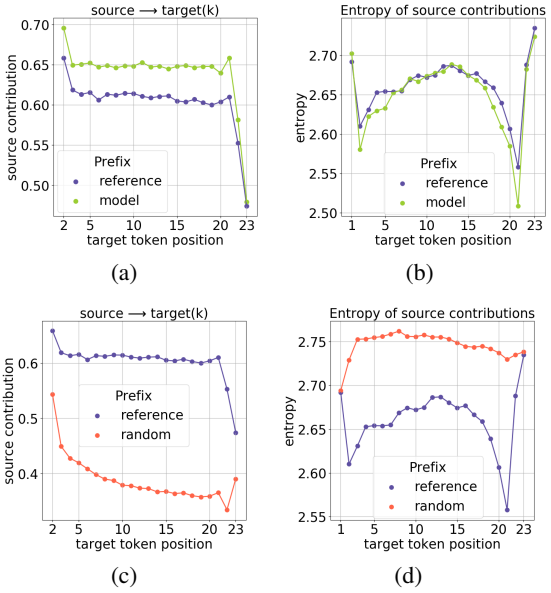
Figure 2: For each generation step, the figure shows entropy of (a) source, (b) target contributions.

Source(k) $\longrightarrow$ target. Now we want to understand if there is a tendency to use source tokens at certain positions more than tokens at the others. For each source token position k, we evaluate its total contribution to the whole target sequence. To eliminate the effect of decreasing source influence during generation, at each step t we normalize source contributions $\mathrm{R}_t(x_k)$ over the total contribution of source at this step $\mathrm{R}_t(\text{source})$. Formally, for the k-th token we evaluate $\sum_{t=1}^{T} \mathrm{R}_t(x_k)/\mathrm{R}_t(\text{source})$. For convenience, we multiply the result by $\frac{S}{T}$: this makes the average total contribution of each token equal to 1.

Figure 1(b) shows that the end-of-sentence token is used much more actively than the rest; the two tokens before that (i.e., the final punctuation mark and the last content token) also influence translations more than other tokens. We hypothesize that this is because these last tokens are relevant not only for generation of the corresponding target tokens, but also for earlier tokens. For example, they may be used to account for the distance to the end of sentence or to understand the tone for the whole sentence. Except for these last three tokens, source tokens at earlier positions influence translations more than tokens at later ones. This may be because the alignment between English and French languages is roughly monotonic. We leave for future work investigating the changes in this behavior for language pairs with more complex alignment (e.g., English-Japanese).

4.2 Entropy of contributions

Now let us look at how 'sharp' contributions of source or target tokens are at different generation steps. For each step t, we evaluate entropy of (normalized) source or target contributions: $\{\mathrm{R}_t(x_i)/\mathrm{R}_t(\text{source})\}_{i=1}^{S}$ or $\{\mathrm{R}_t(y_j)/\mathrm{R}_t(\text{target})\}_{j=1}^{t-1}$.



(a) (b)

(c) (d)

Figure 3: (a, c) contribution of source, (b, d) entropy of source contributions.

Entropy of source contributions. Figure 2(a) shows that during generation, entropy increases until approximately $2/3$ of the translation is generated, then decreases when generating the remaining part. Interestingly, for the last punctuation mark and the EOS token, entropy of source contributions is very high: the decision to complete the sentence requires broader context.

Entropy of target contributions. Figure 2(b) shows that entropy of target contributions is higher for longer prefixes. This means that the model does use longer contexts in a non-trivial way.

4.3 Reference, Model and Random Prefixes

Let us now look at how model behavior changes when feeding different types of prefixes: prefixes of reference translations, translations generated by the model, and random sentences in the target language.[7] As in previous experiments, we evaluate relevance for top-1 logit predicted by the model.

Reference vs model prefixes. When feeding model-generated prefixes, the model uses source more (Figure 3(a)) and has more focused source contributions (lower entropy in Figure 3(b)) than when generating the reference. This may be because model-generated translations are 'easier' than references. For example, beam search translations contain fewer rare tokens (Burlot and

[7] Random prefixes come from the same evaluation set, but with shuffled target sentences.

Yvon, 2018; Ott et al., 2018), are simpler syntactically (Burlot and Yvon, 2018) and, according to the fuzzy reordering score (Talbot et al., 2011), model translations have significantly less reordering compared to the real parallel sentences (Zhou et al., 2020). As we see from our experiments, these simpler model-generated prefixes allow for the model to rely on the source more and to be more confident when choosing relevant source tokens.

Reference vs random prefixes. Results for random sentence prefixes are given in Figures 3c, 3d. The reaction to random prefixes helps us study the self-recovery ability of NMT models. Previous work has found that models can fall into a hallucination mode where "the decoder ignores context from the encoder and samples from its language mode" (Koehn and Knowles, 2017; Lee et al., 2018). In contrast, He et al. (2019b) found that a language model is able to recover from artificially distorted history input and generate reasonable samples.

Our results show that the model tends to fall into hallucination mode even when a random prefix is very short, e.g. one token: we see a large drop of source influence for all positions (Figure 3c). Figure 3d also shows that with a random prefix, the entropy of source contributions is high and is roughly constant.

5 Exposure Bias and Source Contributions

The results in the previous section agree with some observations made in previous work studying self-recovery and hallucinations. In this section, we illustrate more explicitly how our methodology can be used to shed light on the effects of exposure bias and training objectives.

Wang and Sennrich (2020) empirically link the hallucination mode to exposure bias (Ranzato et al., 2016), i.e. the mismatch between the gold history seen at training time, and the (potentially erroneous) model-generated prefixes at test time. The authors hypothesize that exposure bias leads to an over-reliance on target history, and show that Minimum Risk Training (MRT), which does not suffer from exposure bias, reduces hallucinations. However, they did not directly measure this over-reliance on target history. Our method is able to directly test whether there is indeed an over-reliance on the target history with MLE-trained models, and more robust inclusion of source context with MRT. We also consider a simpler heuristic, word dropout, which we hypothesize to have a similar effect.

Minimum Risk Training (Shen et al., 2016) is a sentence-level objective that inherently avoids exposure bias. It minimises the expected loss ('risk') with respect to the posterior distribution:

$$\mathcal{R}(\theta) = \sum_{(x,y)} \sum_{\tilde{y} \in \mathcal{Y}(x)} P(\tilde{y}|x,\theta)\Delta(\tilde{y},y),$$

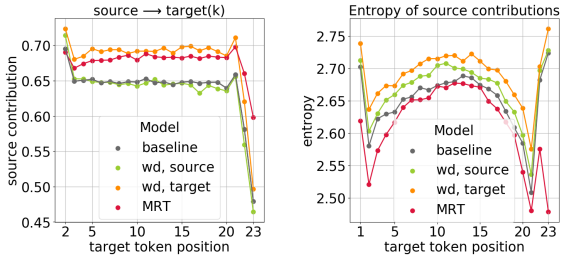
where $\mathcal{Y}(x)$ is a set of candidate translations for x, $\Delta(\tilde{y},y)$ is the discrepancy between the model prediction $\tilde{y}$ and the gold translation y (e.g., a negative smoothed sentence-level BLEU). More details on the method can be found in Shen et al. (2016) or Edunov et al. (2018); training details for our models are in the appendix.

Word Dropout is a simple data augmentation technique. During training, it replaces some of the tokens with a special token (e.g., UNK) or a random token (in our experiments, we replace 10% of the tokens with random). When used on the target side, it may serve as the simplest way to alleviate exposure bias: it exposes a model to something other than gold prefixes. This is not true when used on the source side, but for analysis, we consider both variants.

5.1 Experiments

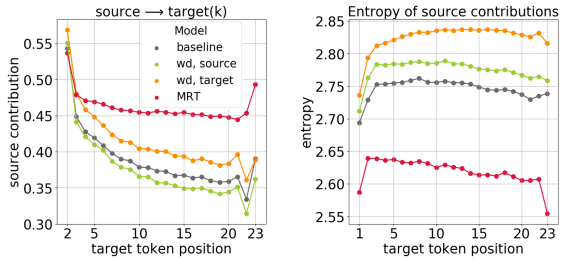
We consider two types of prefixes: model-generated and random. Random prefixes are our main interest here. We feed prefixes that are fluent but unrelated to the source and look whether a model is likely to fall into a language modeling regime, i.e., to what extent it ignores the source. For model-generated prefixes, we do not expect to see large differences in contributions: this mode is 'easy' for the model and the source contributions are high (see Section 4.3). The results are shown in Figures 4 and 5.

Model-generated prefixes. We see that both MRT and target-side word dropout increase influence or the source, while source-side word dropout does not lead to noticeable changes in the total source contribution (Figure 4). This agrees with our hypothesis that exposure bias leads to higher reliance on the target. Results for the entropy of source contributions show that only MRT makes the model more confident in the choice of relevant source tokens, while both variants of word dropout force it to rely on broader context (Figure 4b).



(a) (b)

Figure 4: Contribution of source (a) and entropy of source contributions (b) with model-generated prefixes.



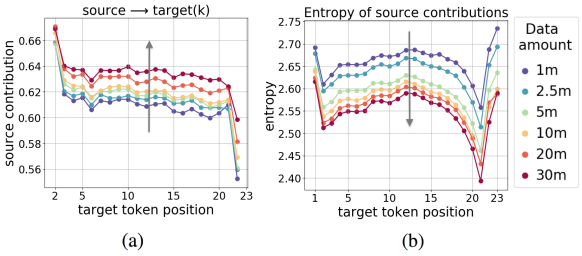
(a) (b)

Figure 5: Contribution of source (a) and entropy of source contributions (b) with random prefixes.

Random prefixes. We see that, among all models, the MRT model has the highest influence of source (Figure 5a) and the most focused source contributions (Figure 5b). This agrees with our expectations: by construction, MRT removes exposure bias completely. Therefore, it is confused by random prefixes less than other models. Additionally, this also links to Wang and Sennrich (2020) who showed that MRT reduces hallucinations.

When using word dropout, the target-side variant increases the influence of source (i.e. decreases the influence of target), while the source-side variant decreases (Figure 5a). This is expected: replacing some words with random decreases model's reliance on the corresponding part of input (either the source or the prefix). Note also that target-side word dropout slightly reduces exposure bias (in contrast to source-side word dropout), which gives us another piece of evidence that reducing exposure bias increases the influence of the source.

Experiments in this section highlight that the methodology we propose can be applied to study exposure bias, robustness, and hallucinations, both in machine translation and more broadly for other language generation tasks. In this work, however, we want to illustrate more broadly the potential of this approach. In the following, we will compare models trained with varying amounts of data and will look into the training process.



(a) (b)

Figure 6: (a) source contribution, (b) entropy of source contributions. The arrows show the direction of change when increasing data amount. (For clarity, in (a) the last two positions (punct. and EOS) are not shown).

6 Data Amount

In this section, we show how the results from Section 4 change when increasing the amount of training data. The observed patterns are the same when evaluating on datasets with reference translations or the ones generated by the corresponding model (in each case, all sentences in the evaluation dataset have the same length). In the main text, we show figures for references.

More data $\Longrightarrow$ higher source contribution. Figure 6(a) shows the source contribution at each generation step. We can see that, generally, models trained with more data rely on source more heavily.

More data $\Longrightarrow$ more focused contributions. Figure 6(b) shows that at each generation step, entropy of source contributions decreases with more data. This means that with more training data, the model becomes more confident in the choice of important tokens.

7 Training Stages

Now we turn to analyzing the training process of an NMT model. Specifically, we look at the changes in how the predictions are formed: changes in the amount of source/target contributions and in the entropy of these contributions. Our findings are summarized in Figure 7. In the following, we explain them in more detail. In Section 7.1, we draw connections between our training stages (shown in Figure 7) and the ones found in previous work focused on validating the lottery ticket hypothesis.

Contributions converge early. First, we evaluate how fast the contributions converge, i.e., how quickly a model understands which tokens are the most important for prediction. For

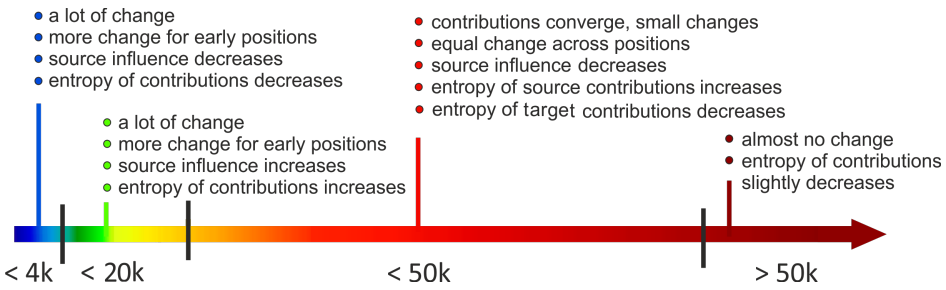
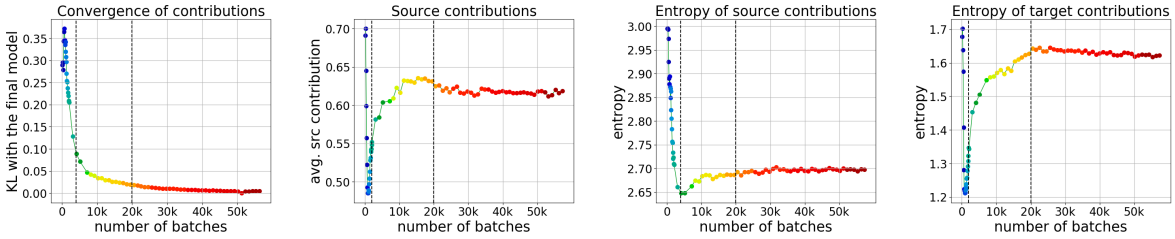


Figure 7: Training timeline.



(a) (b) (c) (d)

Figure 8: Training process: (a) convergence of contributions, (b) source contribution, (c-d) entropy of source and target contributions. The model trained on 1m subsample of WMT14 En-Fr dataset. The results are averaged over target positions and evaluation examples.

this, at each generation step t we evaluate the KL divergence in token influence distributions $(\mathrm{R}_t(x_1), \ldots, \mathrm{R}_t(x_S), \mathrm{R}_t(y_1), \ldots, \mathrm{R}_t(y_{t-1}))$ from the final converged model to the model in training. Figure 8(a) shows that contributions converge early. After approximately 20k batches, the model is very close to its final state in the choice of tokens to rely on for a prediction.

Changes in training are not monotonic. Figures 8(b-d) show how the amount of source contribution and the entropy of source and target contributions change in training. We see that all three figures have the same distinct stages (shown with vertical lines). First, source influence decreases, and both source and target contributions become more focused. In this stage, most of the change happens (Figure 8(a)). In the second stage, the model also undergoes substantial change, but all processes change their direction: source influence increases and the model learns to rely on broader context (entropy is increasing). Finally, in the third stage, the direction changes again for the total source contribution and the entropy of target contributions, and remains the same for the entropy of source contributions. However, very little is going on – the model slowly converges.

These three stages correspond to the first three stages shown in Figure 7; at this point, the model trained on 1m sentence pairs converges. With more data (e.g., 20m sentence pairs), we further observed the next stage (the last one in Figure 7), where the entropy of both source and target contributions is decreasing again. However, this last stage is much slower than the third, and the final state does not differ much from the end of the third stage.

Early positions change more. Figure 9 shows how entropy of source contributions changes for each target position. We see that earlier positions are the ones that change most actively: at these positions, we see the largest decrease at the first stage and the largest following increase at the subsequent stages. If we look at how accuracy for each position changes in training (Figure 10), we see that at the end of the first stage, early tokens have the highest accuracy.[8] This is not surprising: one could expect early positions to train faster because they are observed more frequently in training. Previously such intuition motivated the usage of sentence length as one of the criteria for curriculum learning (e.g., Kocmi and Bojar (2017)).

7.1 Relation to Previous Work

Interestingly, our stages in Figure 7 agree with the ones found by Frankle et al. (2020) for ResNet-20 trained on CIFAR-10 when investigating, among other things, the lottery ticket hypothesis (Frankle

[8] Accuracy is the proportion of cases where the correct token is the most probable choice.

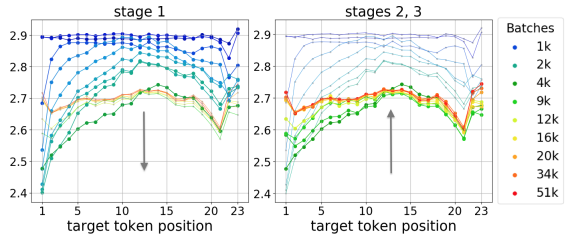


Figure 9: Entropy of source contributions. Changes in training for each target position; each line corresponds to a model state. The arrows show the direction of change when the training progresses. In the figures, all stages are shown, but the stages of interest are highlighted more prominently.

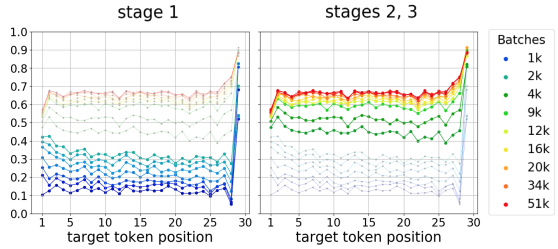


Figure 10: Accuracy change for each target position; each line corresponds to a model state. In the figures, all stages are shown, but the stages of interest are highlighted more prominently.

and Carbin, 2019). Their stages were defined based on the changes in gradient magnitude, in the weight space, in the performance, and in the effectiveness of rewinding in search of the 'winning' subnetwork (for more details on the lottery ticket hypothesis and the rewinding, see the work by Frankle et al. (2019)). Comparing the stages by Frankle et al. (2020) with ours, we see that (1) their relative sizes in the corresponding timelines match well, (2) the rewinding starts to be effective at the third stage; for our model, this is when the contributions have almost converged. In future work, it would be interesting to further investigate this relation.

8 Additional Related Work

To estimate the influence of source to an NMT prediction, Ma et al. (2018) trained an NMT model with an auxiliary second decoder where the encoder context vector was masked. Then the source influence was measured as the KL divergence between predictions of the two decoders. However, the ability of an auxiliary decoder to generate similar distribution is not equivalent to the main model not using source. More recently, as a measure of individual token importance, He et al. (2019a) used Integrated Gradients (Sundararajan et al., 2017).

In machine translation, LRP was previously used for visualization (Ding et al., 2017) and to find the most important attention heads in the Transformer's encoder (Voita et al., 2019). Similar to our work, Voita et al. (2019) evaluated LRP on average over a dataset (and not for a single prediction) to extract patterns in model behaviour. Both works used the more popular ε-LRP, while for our analysis, the $\alpha\beta$-LRP was more suitable (Section 2). For language modeling, Calvillo and Crocker (2018) use LRP to evaluate relevance of neurons in RNNs for a small synthetic setting.

9 Conclusions

We show how to use LRP to evaluate the relative contributions of source and target to NMT predictions. We illustrate the potential of this approach by analyzing changes in these contributions when conditioning on different types of prefixes (references, model predictions or random translations), when varying training objectives or the amount of training data, and during the training process. Some of our findings are: (1) models trained with more data rely on source more and have more sharp token contributions; (2) the training process is non-monotonic with several distinct stages. These stages agree with the ones found in previous work focused on validating the lottery ticket hypothesis, which suggests future investigation of this connection. Additionally, we show that models suffering from exposure bias are more prone to over-relying on target history (and hence to hallucinating) than the ones where the exposure bias is mitigated. In future work, our methodology can be used to measure the effects of different and novel training regimes on the balance of source and target contributions.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. We also thank Wenxu Li for noticing an error in the first version of our released code. The work is partially supported by the European Research Council (Titov, ERC StG BroadSem 678254), Dutch NWO (Titov, VIDI 639.022.518) and EU Horizon 2020 (GoURMET, no. 825299). Lena is supported by the Facebook PhD Fellowship. Rico Sennrich acknowledges support of the Swiss National Science Foundation (MUTAMUR; no. 176727).

References

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

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.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*, New Orleans.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. *Lecture Notes in Computer Science*, page 63–71.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Jesús Calvillo and Matthew Crocker. 2018. Language production dynamics with recurrent neural networks. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 17–26, Melbourne. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2019. Stabilizing the lottery ticket hypothesis.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. 2020. The early phase of neural network training. In *International Conference on Learning Representations*.

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. Asian Federation of Natural Language Processing.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019a. Towards understanding neural machine translation with word importance. 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 953–962, Hong Kong, China. Association for Computational Linguistics.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. 2019b. Quantifying exposure bias for neural language generation.

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

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation.

Xintong Li, Lemao Liu, Rui Wang, Guoping Huang, and Max Meng. 2020. Regularized context gates on transformer for machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Xutai Ma, Ke Li, and Philipp Koehn. 2018. An analysis of source context dependency in neural machine translation.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. Layer-wise relevance propagation: an overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 193–209. Springer.

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.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

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.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia. PMLR.

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.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. Association for Computational Linguistics.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Belgium, Brussels. Association for Computational Linguistics.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

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, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, Deyi Xiong, and Chao Bian. 2018. Neural machine translation with decoding history enhanced attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1464–1473, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

A Experimental setup

A.1 Data preprocessing

We use random subsets of the WMT14 En-Fr dataset: http://www.statmt.org/wmt14/translation-task.html. Sentences were encoded using byte-pair encoding (Sennrich et al., 2016), with source and target vocabularies of about 32000 tokens. Translation pairs were batched together by approximate sequence length. Each training batch contained a set of translation pairs containing approximately 16000[9] source tokens for 1m subsample and 32000 for larger datasets.

A.2 Model parameters

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).

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

We train models till convergence and average 5 latest checkpoints. Approximate number of training batches are: 57k for 1m dataset, 220k for 2.5m dataset and 600k for the rest.

B Minimum Risk Training

B.1 Background

Minimum Risk Training (MRT) minimises the expected loss ('risk') with respect to the posterior distribution:

$$\mathcal{R}(\theta) = \sum_{(x,y)} \sum_{\tilde{y} \in \mathcal{Y}(x)} P(\tilde{y}|x,\theta)\Delta(\tilde{y},y),$$

where $\mathcal{Y}(x)$ is a set of all possible candidate translations for x, $\Delta(\tilde{y},y)$ is the discrepancy between the model prediction $\tilde{y}$ and the gold translation y.

Since the search space $\mathcal{Y}(x)$ is exponential, in practice it is common to use only a subset of the full space. Formally, instead of $\mathcal{Y}(x)$ we use $\mathcal{S}(x) \in \mathcal{Y}(x)$, where $\mathcal{S}(x)$ is obtained by sampling several translations. The probabilities $P(\tilde{y}|x,\theta)$ are replaced with the $\tilde{P}$, which is renormalized over the subset $\mathcal{S}$:

$$\tilde{P}(\tilde{y}|x,\theta,\alpha) = \frac{P(\tilde{y}|x,\theta)^\alpha}{\sum_{y' \in \mathcal{S}(x)} P(y'|x,\theta)^\alpha}.$$

The hyperparameter α is used to control the sharpness of the distribution.

B.2 Experimental setting

To choose the setting, we mostly relied on previous work (Shen et al., 2016; Edunov et al., 2018). Model is pre-trained with the token-level objective MLE and then fine-tuned with MRT; the fine-tuning stage is approximately one epoch.

Candidate translations. The translations are sampled using standard random sampling without temperature. Following Shen et al. (2016), we take the large number of candidates; specifically, we use 50 translations and add a reference to the subset. While Edunov et al. (2018) report that adding the reference to the set of candidates hurts quality, in preliminary experiments we found that this was not the case for our setting.

Measure of discrepancy. The measure of discrepancy, $\Delta(\tilde{y},y)$, is a negative smoothed sentence-level BLEU.

Batch size. On average, the number of examples (where an example is a translation pair along with all candidates) is the same as in training of the baseline models. This is achieved by accumulating gradients for several steps and making an update.

Other parameters. Following (Wang and Sennrich, 2020), we set $\alpha = 0.005$ and the learning rate to 0.00001.

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