

Is Encoder-Decoder Redundant for Neural Machine Translation?

Yingbo Gao Christian Herold Zijian Yang Hermann Ney

Human Language Technology and Pattern Recognition Group

Computer Science Department

RWTH Aachen University

D-52056 Aachen, Germany

{ygao|herold|zyang|ney}@cs.rwth-aachen.de

Abstract

Encoder-decoder architecture is widely adopted for sequence-to-sequence modeling tasks. For machine translation, despite the evolution from long short-term memory networks to Transformer networks, plus the introduction and development of attention mechanism, encoder-decoder is still the de facto neural network architecture for state-of-the-art models. While the motivation for decoding information from some hidden space is straightforward, the strict separation of the encoding and decoding steps into an encoder and a decoder in the model architecture is not necessarily a must. Compared to the task of autoregressive language modeling in the target language, machine translation simply has an additional source sentence as context. Given the fact that neural language models nowadays can already handle rather long contexts in the target language, it is natural to ask whether simply concatenating the source and target sentences and training a language model to do translation would work. In this work, we investigate the aforementioned concept for machine translation. Specifically, we experiment with bilingual translation, translation with additional target monolingual data, and multilingual translation. In all cases, this alternative approach performs on par with the baseline encoder-decoder Transformer, suggesting that an encoder-decoder architecture might be redundant for neural machine translation.

1 Introduction

Sequence-to-sequence modeling is often approached with Neural Networks (NNs), prominently encoder-decoder NNs, nowadays. For the task of Machine Translation (MT), which is by definition also a sequence-to-sequence task, the default choice of NN topology is also an encoder-decoder architecture. For example, in early works like Kalchbrenner and Blunsom (2013), the authors already make the distinction between their convolutional sentence model (encoder) and recurrent

language model (decoder) conditioned on the former. In follow-up works like Sutskever et al. (2014) and Cho et al. (2014a,b), the concept of encoder-decoder network is further developed. While extensions such as attention (Bahdanau et al., 2014), multi-task learning (Luong et al., 2015), convolutional networks (Gehring et al., 2017) and self-attention (Vaswani et al., 2017) are considered for sequence-to-sequence learning, the idea of encoding information into some hidden space and decoding from that hidden representation sticks around.

Given the success and wide popularity of the Transformer network (Vaswani et al., 2017), many works focus on understanding and improving individual components, e.g. positional encoding (Shaw et al., 2018), multi-head attention (Voita et al., 2019), and an alignment interpretation of cross attention (Alkhouli et al., 2018). In works that go a bit further and make bigger changes in terms of modeling, e.g. performing round-trip translation (Tu et al., 2017) and going from autoregressive to non-autoregressive (Gu et al., 2017), the encoder-decoder setup itself is not really questioned. In the mean time, it is not to say that the field is completely dominated by one approach. Because works like the development of direct neural hidden Markov model (Wang et al., 2017, 2018, 2021b), investigation into dropping attention and separate encoding and decoding steps (Press and Smith, 2018) and going completely encoder-free (Tang et al., 2019) do exist, where the default encoder-decoder regime is not directly applied.

Meanwhile, in the field of language modeling, significant progress is achieved with the wide application of NNs. With the progress from early feedforward language models (LMs) (Bengio et al., 2000), to the successful long short-term memory network LMs (Sundermeyer et al., 2012), and to the more recent Transformer LMs (Irie et al., 2019), the modeling capacity of LMs nowadays is much more than their historic counterparts. This is es-

pecially true when considering some of the most recent extensions, such as large-scale modeling (Brown et al., 2020), modeling very long context (Dai et al., 2019) and going from autoregressive modeling to non-autoregressive modeling (Devlin et al., 2019). Because MT can be thought of as a contextualized language modeling task with the source sentence being additional context, one natural question is if simply concatenating the source and target sentences and train an LM to do translation would work (Irie, 2020). This idea is simple and straightforward, but special care needs to be taken about the attention mechanism and source reconstruction. In this work, we explore this alternative approach and conduct experiments in bilingual translation, translation with additional target monolingual data and multilingual translation. Our results show that dropping the encoder-decoder architecture and simply treating the task of MT as contextualized language modeling is sufficient to obtain state-of-the-art results in translation. This result has several subtleties and implications, which we discuss in Sec.5, and opens up possibilities for more general interfaces for multimodal modeling.

2 Related Work

In the literature, few but interesting works exist which closely relate to the idea mentioned above. In Mikolov and Zweig (2012), the authors mention the possibility to use source sentence as context for contextualized language modeling. In He et al. (2018), with the intuition to coordinate the learning of Transformer encoder and decoder layer by layer, the authors share the encoder and decoder parameters and learn a joint model on concatenated source and target sentences. However, no explicit source side reconstruction loss is included. Similarly, in Irie (2020), a small degradation in translation quality is observed when a causal mask is used and no source reconstruction is included. Because the masking is critical for correctly modeling the dependencies regarding the concatenated sequence, in Raffel et al. (2020), the authors put special focus on discussing the differences and implications of three types of attention masks. In Wang et al. (2021a), the authors expand upon the idea and propose a two-step decaying learning rate schedule to reconstruct the source sentence to regularize the training process. In that work, the authors show competitive performance compared to Transformer baselines in several settings. More recently, in

Zhang et al. (2022), the authors also use a language-modeling-style source side reconstruction loss to regularize the model, and additionally explore the model scaling cross-lingual transfer capabilities. Another work that explores the long-context modeling potential of LMs is Hawthorne et al. (2022), where data from domains other than translation is included in model training. Hao et al. (2022) is a more recent addition to this direction of research, where LM as a general interface for multimodal data is investigated. Because our focus is in MT, we refer to such a model, where encoder-decoder architecture is dropped and an LM is used to model the concatenation of source and target sentence, as Translation Language Models (TLMs¹).

The work by Wang et al. (2021a) is probably the most directly related work compared to our work, therefore we believe it is important to highlight the similarities and differences between their work and ours. The core concept of dropping encoder-decoder architecture is similar between Wang et al. (2021a) and our work, and competitive performance of TLMs compared to encoder-decoder models in various settings is achieved in both works. However, we additionally explore the task of autoencoding in the source side, adding Bidirectional-Encoder-Representations-from-Transformers-style (BERT) noise (Devlin et al., 2019), using alternative learning rate schedules, training MT models with back-translated (BT) data and doing multilingual training. Further, we discuss subtleties and implications associated with the TLM.

3 Methodology

The core concept of TLM is to concatenate the source and the target sentences and treat the translation task as a language modeling task during training. The two majors points of concern are the attention mechanism and the source-side reconstruction loss. In this section, we explain the details related to these two points, and additionally discuss the implications when additional target-side monolingual data or multilingual data is available.

3.1 Translation Language Model

Denoting the source words/subwords as f and the target words/subwords as e , with running indices

¹To be differentiated from TLMs in Conneau and Lample (2019), where the pretraining objective is cloze task at both source and target side, using bilingual context.

j in J and i in I respectively, the usual way to approach the translation problem in encoder-decoder models is to directly model the posterior probabilities via a discriminative model $P(e_1^I|f_1^J)$. This is used in the Transformer and can be expressed as:

$$P(e_1^I|f_1^J) = \prod_{i=1}^I P(e_i|e_0^{i-1}, f_1^J).$$

The model is usually trained with the cross entropy criterion (often regularized with label smoothing (Gao et al., 2020b)), and the search aims to find the target sentence \hat{e}_1^I with the highest probability (often approximated with beam search):

$$L_{\text{MT}} = - \sum_{i=1}^I \log P(e_i|e_0^{i-1}, f_1^J),$$

$$\hat{e}_1^I = \arg \max_{e_1^I} \{\log P(e_1^I|f_1^J)\}.$$

Alternatively, one can model the joint probability of the source and target sentences via a generative model $P(f_1^J, e_1^I)$ and it can be expressed as:

$$P(f_1^J, e_1^I) = \prod_{j=1}^J P(f_j|f_0^{j-1}) \prod_{i=1}^I P(e_i|e_0^{i-1}, f_1^J).$$

Here, because f_1^J is given at search time, and $\arg \max_{e_1^I} P(f_1^J, e_1^I) = \arg \max_{e_1^I} P(e_1^I|f_1^J)$, the search stays the same as in the baseline case. But the training criterion has an additional loss term on the source sentence, which we refer to as reconstruction loss (L_{RE}), the learning rate λ of which can be controlled by some schedule:

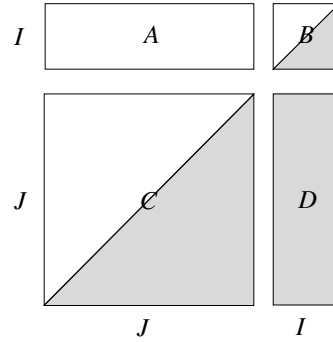
$$L_{\text{RE}} = - \sum_{j=1}^J \log P(f_j|f_0^{j-1}),$$

$$L_{\text{TLM}} = \lambda L_{\text{RE}} + L_{\text{MT}}.$$

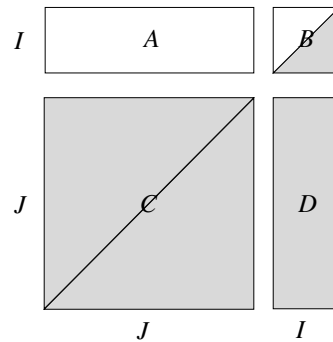
One can think of the reconstruction loss (decomposed in an autoregressive manner here, but it does not have to be) as a second task in addition to the translation task, or simply a regularization term for better learning of the source hidden representations. Although this formulation is simple and straightforward, there could be variations in how the source side dependencies are defined.

3.1.1 On the Attention Mechanism

In the original Transformer (Vaswani et al., 2017) model, the attention mechanism is used in three



(a) source-side triangular mask



(b) source-side full mask

Figure 1: Attention masks in TLM with (a) a triangular mask, and (b) a full mask, at the source side. The horizontal direction is the query direction and the vertical direction is the key direction. Shaded areas mean that the attention is valid and white areas mean that the attention is blocked. The matrices C, B, and D correspond to the encoder self attention, the decoder self attention and encoder-decoder cross attention in Transformer, respectively. The matrix A is whitened in both cases because we should not allow the source positions attend to future target positions.

places, namely, a $J \times J$ encoder self attention matrix, a $I \times I$ decoder self attention matrix and a $J \times I$ encoder-decoder cross attention matrix. As shown in Fig.1, they correspond to matrices C, B and D respectively. The attention masks in B and D are straightforward. The triangular attention mask in the B matrix needs to be causal by definition, because otherwise target positions may attend to future positions and cheat. The attention mask in D needs to be full, because we want each target position to be able to look at each source position so that there is no information loss. However, the attention mask in C is how some of the previous works differ. For example, a triangular attention mask like in Fig.1a is used in Irie (2020), while a full attention mask like in Fig.1b is used in He et al.

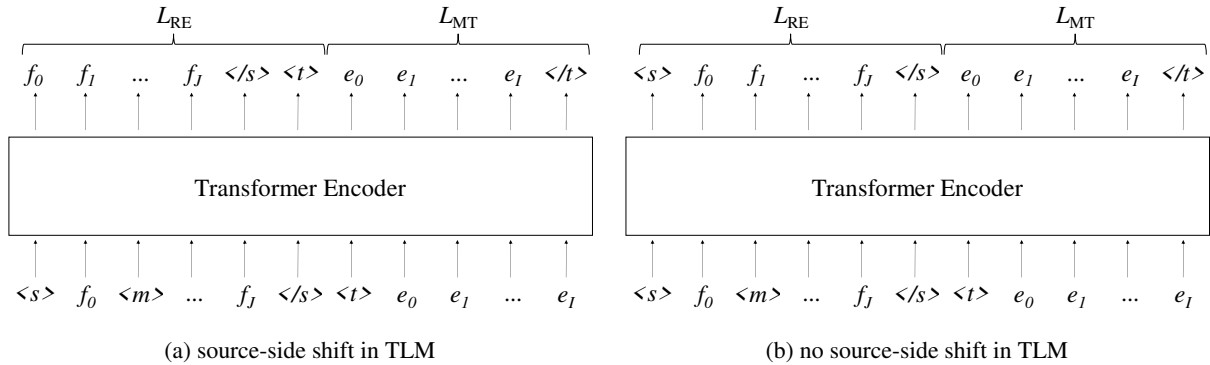


Figure 2: Shifting versus no shifting of the output at the source side in TLM. The output at the target side is shifted in both cases. $\langle s \rangle$, $\langle /s \rangle$, $\langle t \rangle$ and $\langle /t \rangle$ are artificial start and end of sentence symbols at the source and target side respectively². $\langle m \rangle$ denotes BERT-style (Devlin et al., 2019) randomly masked tokens. When matrix C in Fig.1 is triangular, (a) corresponds to a language modeling objective. When C is full, (b) corresponds to an auto-encoding objective. During search, $\langle s \rangle, f_0, \dots, f_J, \langle /s \rangle, \langle t \rangle$ is presented to the model, and beam search is done by minimizing L_{MT} .

(2018). Raffel et al. (2020) and Zhang et al. (2022) also discuss the differences in masking patterns in the matrix C similar to what we do here. Wang et al. (2021a) do not make clear what type of attention masks is used in C in their paper, and we do not find a public repository associated with their paper to further investigate it.

In our case, we consider both the triangular and full attention mask patterns for C, because both have good intuitions. The triangular mask is closer to the original objective of learning the joint distribution $P(f_1^J, e_1^I)$, while the full mask enables better information flow because early source positions also have access to future source positions to come up with better hidden representations. That said, later we show through experiments, that for the task of MT, it is clearly better to use a full attention mask for C in TLM.

The matrix A in Fig.1 is whitened throughout this work, because we do not allow the source positions attend to target positions. However, theoretically, when decoding position i , one could allow all source positions $1, 2, \dots, J$ to attend to all previous target positions $1, 2, \dots, i - 1$. This can be done by using a $(J + I) \times (J + I) \times I$ attention mask tensor. The extended I dimension is target-position-dependent, providing a different view of the $(J + I) \times (J + I)$ matrix for each target posi-

²The exact format of the tags is not important so long as uniquely identifiable translation direction tags are used, be it source and target tags like $\langle s \rangle$, $\langle /s \rangle$, $\langle t \rangle$, $\langle /t \rangle$, or direction tags like $\langle s2t \rangle$, or even only the target language tag $\langle t \rangle$. While the later two reduce the total sequence length, the former is more versatile when data from multiple languages or multiple modalities is considered.

tion. Intuitively, this has the potential to serve as an implicit fertility model.

3.1.2 On the Reconstruction Loss

In the paper by Wang et al. (2021a), the source side reconstruction is formulated as an autoregressive language modeling task. However, that does not have to be the case. For example, one can make the distinction to shift or not shift the output at the source side, as shown in Fig.2. When the source output is shifted, L_{RE} is a normal language modeling cross entropy loss. When the source output is not shifted, L_{RE} is an auto-encoding loss. Additionally considering the matrix C in Fig.1, assuming no source input noise is introduced, then when C is full, or when C is triangular but the source output is not shifted, the source-side reconstruction becomes a trivial copying task.

On top of the reconstruction loss formulation, one can also apply noises to the source side input. This can be viewed as a regularization or a data augmentation trick, such that the source side information is corrupted to a certain degree to help the generalization ability of the model. In this work, we consider the BERT-style (Devlin et al., 2019) noises, where 15% of source positions are picked at random, and 80%, 10% and 10% of the tokens in this positions are replaced with $\langle m \rangle$, a random token or unchanged, respectively. Different to the BERT paper though, in addition to the cloze task in the masked positions, we also keep the cross entropy losses in the unmasked positions. One can of course go over the 15% (Wettig et al., 2022) limit or apply softer noises (Gao et al., 2019, 2020a), but

we do not further expand in this direction because it is beyond our initial goal to verify the necessity of the encoder-decoder architecture.

One more thing that can be tuned for the reconstruction loss is the learning rate schedule. In Wang et al. (2021a), a two-step linear decaying function is used, where λ linearly decays to 0.1 until a certain number of gradient update steps τ , and decays with a smaller rate after τ . Here, we additionally consider schedules where the learning rate λ : (a) is constant at zero, (b) is constant at one, (c) two-step linearly decays like in Wang et al. (2021a) and (d) decays exponentially as $\lambda_t = \exp(-\ln 0.1t/\tau)$. Similar to (c), the schedule (d) decays to 0.1 at gradient update step τ as well.

3.2 Bilingual and Monolingual Training

For MT, target-side monolingual data is often available in large quantities and is shown to be helpful for the main task of translation when used in one way or another (Koehn et al., 2007; Wuebker et al., 2012; Freitag et al., 2014; Sennrich et al., 2016a; Gulcehre et al., 2017a; Domhan and Hieber, 2017; Stahlberg et al., 2018; Edunov et al., 2018; Graça et al., 2019). Broadly speaking, they can be categorized into three approaches: 1. ensembling with an external language model, 2. multi-task training with additional language modeling objective and 3. training with back-translated data with artificial source and true target. Evidence so far is that back-translation works the best among the three (Barrault et al., 2021).

For TLM, these three approaches are all applicable, but with implications. First, ensembling is not very relevant because of the additional training and storage requirements, and also it is against the philosophy of TLM where we want to make the encoder-decoder model more compact. Second, the multi-task training is interesting because while some previous work have dedicated layers to perform the language modeling task (Gulcehre et al., 2015, 2017b), such multi-task training on TLM actually trains all model parameters in the auxiliary language modeling task. Third, the back-translation approach is worth looking at because it delivers the best results in encoder-decoder models so far and experiments comparing TLM with the baseline under this setting are necessary to justify whether or not we can throw away the encoder-decoder architecture.

3.3 Multilingual Training

Another important setting where TLM needs to be compared to the baseline encoder-decoder model is when multilingual data is used in training. Broadly speaking, multilingual models can refer to systems that translate in one-to-many, many-to-one, many-to-many, or even source-to-target and target-to-source manners. The major benefits of training multilingual models (Johnson et al., 2017; Aharoni et al., 2019) are: more compact models via shared parameters and transfer/zero-shot learning capabilities due to inherit similarities in some languages. While there exist works that propose to use language-specific sub-networks to take into consideration the parameter capacity needed for each language, e.g. in Lin et al. (2021), it is more common to simply train one joint model where the model parameters are shared across all languages.

For TLM, the task of multilingual training is straightforward as well. One can simply concatenate each translation pair into one longer sequence, add corresponding translation direction tags, and feed the concatenated sequence to the TLM model. In other words, all the hidden parameters of the model can be shared across all translation directions, and one simply needs to pay attention to the word embeddings such that words/subwords/tokens from different languages are mapped into the same embedding space for further processing, similar to what is done for encoder-decoder models.

4 Experiments

To verify the performance of TLM compared to the baseline encoder-decoder Transformer model, we perform experiments on four machine translation datasets. Specifically, we experiment with the International Conference on Spoken Language Translation (IWSLT) (Federico et al., 2014) 2014 German-to-English (de-en), the Conference on Machine Translation (WMT) 2016 English-to-Romanian (en-ro) (Bojar et al., 2016), 2019 Chinese-to-English (zh-en) (Barrault et al., 2019) datasets. Additionally, for multilingual experiments, we create a custom multilingual (multi.) dataset from news-commentary v16 (Tiedemann, 2012), performing translation among three languages, German (de), Spanish (es), and French (fr), in six direction: de-

es, es-de, de-fr, fr-de, es-fr, fr-es³. For the monolingual data, we sample 5M sentences from the English News crawl monolingual corpus⁴. To create synthetic zh-en data, we employ our en-zh Transformer model to do back-translation (Sennrich et al., 2016a). The data is pre-processed with the Byte Pair Encoding (BPE) (Sennrich et al., 2016b) algorithm. We lowercase the text for de-en and for the other language pairs, we leave the original casing as is. The statistics of the datasets are summarized in Tab.1.

dataset	vocab.	train pairs	test pairs
de-en	10k	0.2M	6k
en-ro	20k	0.6M	2k
multi.	32k	1.7M	18k
zh-en	47k	17.0M	4k

Table 1: Statistics of the datasets.

We implement the Transformer model and the TLM model with different options such as using different attention masks, shifting versus not shifting the source output, adding or not adding BERT-style (Devlin et al., 2019) noises and different learning rate schedules, in PyTorch (Paszke et al., 2019). The back-translation and multilingual experiments are done by adding corresponding language tags to the concatenation of source and target sentences.

We follow the training and search hyperparameters as closely as possible to the original Transformer (Vaswani et al., 2017) paper. Note that, when searching with TLM, the entire source sentence until (and including) the target start of sentence $\langle t \rangle$ is fed into the NN. The beam search is then carried out only on the target outputs. We report translation performances in BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores using the MultEval tool from Clark et al. (2011).

4.1 An Encoder-Only Model

First, we consider the necessity of encoder-decoder architecture by comparing our encoder-only TLM with the baseline Transformer model, on de-en and en-ro. Essentially, we perform a grid search over four hyper-parameters regarding the source reconstruction:

³The data is retrieved from <https://data.statmt.org/news-commentary/v16/>. We take all bilingual data in the six directions. For the raw data in each direction, we take the first 3000 lines as test data and the last 3000 lines as development data.

⁴<https://data.statmt.org/news-crawl/>

1. Language modeling (shifting source output, Fig.2a) versus autoencoding (not shifting source output, Fig.2b).
2. Triangular (see Fig.1a) versus full (Fig.1b) attention mask.
3. No source input noise versus BERT-style (Devlin et al., 2019) source input noise (Sec.3.1.2).
4. Constant learning rate λ for L_{RE} at zero or one, or the two-step linear (Wang et al., 2021a) or exponential decay (Sec.3.1.2).

Due to the limited length, we only highlight the interesting points from our observations and append the full grid-search table (Tab.9) in the appendix for the interested reader. For the discussions below, we consider one hyperparameter each time and pick the best set of other hyperparameters from the grid search, to take into considerations of possible correlations among different hyperparameters.

4.1.1 Both Autoencoding and Language Modeling Work

First, we see that both shifting and not shifting the source output at the source side seem to work for TLM. As shown in Tab.2, when picking the best set of other hyperparameters, TLMs trained with either of the auxiliary task can perform on par with the encoder-decoder baseline within $\pm 0.2\%$ absolute BLEU score fluctuations.

arch.	task	de-en	en-ro
enc-dec	-	34.9	26.0
enc-only	LM	34.7	26.2
	AE	35.0	26.0

Table 2: BLEU scores of language modeling (LM) versus autoencoding (AE) at the source side.

4.1.2 Full Attention Over Source Is Necessary

arch.	mask	de-en	en-ro
enc-dec	-	34.9	26.0
enc-only	triangular	34.4	25.6
	full	35.0	26.2

Table 3: BLEU scores of triangular versus full attention mask at the source side.

Looking at the source attention mask (Tab.3), it is clear that a triangular leads to degradation in translation performance. One interesting setup is

when the attention mask is triangular but the task is autoencoding, i.e. no shift in source outputs. One may argue that the model is allowed to cheat on the auxiliary task L_{RE} because the diagonals in the attention mask is not masked out, however, for the translation task, it is possible that source hidden representations learned from being able to look at future source positions is more beneficial.

4.1.3 BERT-Style Noise Is Slightly Helpful

Moving on to the source-side noise, adding BERT-style (Devlin et al., 2019) seems to slightly boost the translation performance. This observation agrees with past experiences where augmenting the training data with artificial noise regularizes the model for better generalization (Hill et al., 2016; Kim et al., 2018, 2019; Gao et al., 2019, 2020a).

arch.	noise	de-en	en-ro
enc-dec	-	34.9	26.0
enc-only	none	34.6	26.1
	BERT	35.0	26.2

Table 4: BLEU scores with and without BERT-style (Devlin et al., 2019) noises at the source side.

4.1.4 Loss Schedule Is Not Critical

Contrary to Wang et al. (2021a) and also to our surprise, the learning rate schedule for λ does not seem to be critical for obtaining good translation performance with TLM. As shown in Tab.5, even without the reconstruction loss L_{RE} , i.e. when λ is constant at zero, the BLEU score of the TLM is still comparable with the baseline transformer. Of course one needs to tune the other hyperparameters, it is still interesting that the model is able to learn decent source hidden representations even without any auxiliary training signal.

arch.	schedule	de-en	en-ro
enc-dec	-	34.9	26.0
enc-only	0	34.9	26.2
	1	34.5	26.0
	lin	34.7	25.8
	exp	34.8	26.1

Table 5: BLEU scores with different learning rate schedules of λ . "lin" refers to the two-step learning rate decay in Wang et al. (2021a) and "exp" refers to the exponential decay introduced in Sec.3.1.2.

4.1.5 Parameter Count Needs to Be the Same

Although the hyperparameters mentioned so far have different degrees of influence on the final BLEU score, one hyperparameter that governs the overall performance of TLM is the total learnable parameter count. Similar to Wang et al. (2021a), the encoder-only model needs to have a similar amount of parameters to reach the performance of the Transformer baseline. Here, we vary the number of Transformer encoder layers in TLM and compare with the baseline Transformer to illustrate this point. An autoencoding loss is used without shifting the source outputs, noises are added to the source inputs, and a fixed $\lambda = 1$ is used for the encoder-only TLMs in Tab.6. It can be seen that, when the TLM is under- or over- parametrized, underfitting and overfitting happens respectively, leading to worse performances.

arch.	#layers	#params	de-en	
			BLEU	TER
enc-dec	6-6	36.9M	34.9	44.5
enc-only	5	15.9M	33.5	46.2
	10	26.4M	34.9	44.6
	15	36.9M	35.0	44.7
	20	47.4M	34.8	45.1

Table 6: BLEU and TER scores of models of different sizes. For the encoder-decoder model, 6-6 means 6 encoder layers and 6 decoder layers.

arch.	devPPL	zh-en	
		BLEU	TER
enc-dec	6.91	23.2	60.5
+ back-translation	6.21	24.6	59.4
enc-only	6.90	23.1	60.5
+ LM	6.70	23.0	61.4
+ back-translation	6.18	24.7	59.4

Table 7: Transformer versus TLM, with and without additional monolingual target side data.

4.2 Bilingual and Monolingual Training

The streamlined architecture of TLM allows us to easily include monolingual data during training, without the need to create synthetic parallel data and without having to modify the architecture in any way. The system is simply trained jointly on the translation and language modeling tasks. We compare this training strategy to the most common way of including monolingual data in MT train-

arch.	devPPL	de-es	es-de	de-fr	fr-de	es-fr	fr-es	overall
enc-dec	6.17	25.7	19.1	21.3	16.9	24.6	26.2	22.5
enc-only	6.06	25.5	18.8	20.7	16.6	24.4	26.0	22.3

Table 8: BLEU scores of multilingual translation with encoder-decoder Transformer and encoder-only TLM. Here, we train both the encoder-decoder baseline model as well as the encoder-only TLM until the same number of steps and pick the best checkpoint according to the best development set perplexity. The overall score is calculated over the concatenation of the test sets and is not the average of the previous columns.

ing, namely back-translation and experiment on the high resource zh-en task. The results are shown in Tab.7.

As expected, the additional synthetic data from back-translation leads to an improvement in both, development set perplexity (devPPL) and translation quality, for the Transformer and TLM. Including the monolingual data directly in the TLM does also improve perplexity, but does not improve overall translation quality.

4.3 Multilingual Training

The experimental results for the multilingual translation are summarized in Tab.8. Although the encoder-only TLM actually delivers better devPPL than the encoder-decoder Transformer baseline, the BLEU scores are slightly worse (about -0.2% absolute BLEU) across the board. This mismatch between the development set perplexity and the test BLEU in NMT is also reported in previous work (Gao et al., 2020b). We believe this small difference is within acceptable noise range and conclude that the TLM is also on par with the baseline encoder-decoder model in multilingual translation.

5 Discussions

Through extensive experiments, we show that the encoder-decoder architecture is not a must to achieve decent translation performance, because an encoder-only TLM is also capable of obtaining comparable performance when carefully tuned. Here, we touch upon several important implications and subtleties that come with using TLMs.

First, although the encoder-decoder architecture is dropped, the cross attention is still existent in the TLM. As shown in Fig.1, the difference compared to the baseline is that for each target position i , the softmax needs to normalize the attention weights over $J + i$ instead of J . However, because we know the softmax is decent at zeroing out certain positions, e.g. see Fig.1 in Alkhouli et al. (2018),

this should not be a problem. Next, although we do not expand on search in this paper, our internal experiments verify that the search with TLM behaves similarly to the baseline. Further, one may wonder how separate source and target vocabularies should be handled in case of TLMs. Here, we note that having separate source and target word embedding matrices is the same as concatenating them in the vocabulary size dimension into a bigger word embedding matrix for TLM. What could pose as a problem is the increased length of the concatenated sequence. This puts extra requirements to the model and its capabilities to model long context dependencies. Note that, concatenation may not be the only way to combine the source and target contexts. For instance, in the eager model proposed in Press and Smith (2018), the authors essentially "stack" instead of "concatenate". Moreover, when decoding efficiency is critical, TLM may suffer because a separate decoder is not existent and each translation query goes through the entire network. Another limitation is that the source side reconstruction loss considered in this work may also be applied to the Transformer baseline, and might change the picture when comparing the two. That said, TLMs are undoubtedly exciting models opening new possibilities. For example, with such generative models, generation of synthetic translation pairs from scratch can be easily done. Another worth-to-mention application is end-to-end speech translation (ST). While previous work, e.g. in Bahar et al. (2021), connects the encoder of the automatic speech recognition model and the decoder of the MT model, effectively throwing away 50% of the pre-trained model parameters, TLMs can retain all pre-trained parameters and result in more compact end-to-end ST models.

6 Conclusion

In this work, we question the long-standing encoder-decoder architecture for neural machine translation. Through extensive experiments in

various translation directions, considering back-translation and multilingual translation, we find that an encoder-only model can perform as good as an encoder-decoder model. We further discuss implications and subtleties of such models to motivate further research into more compact models and more general neural network interfaces.

Acknowledgements

This work was partially supported by the project HYKIST funded by the German Federal Ministry of Health on the basis of a decision of the German Federal Parliament (Bundestag) under funding ID ZMVI1-2520DAT04A, and by NeuroSys which, as part of the initiative “Clusters4Future”, is funded by the Federal Ministry of Education and Research BMBF (03ZU1106DA).

References

- Roei Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. [On the alignment problem in multi-head attention-based neural machine translation](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium. Association for Computational Linguistics.
- Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. 2021. Tight integrated end-to-end training for cascaded speech translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 950–957. IEEE.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors. 2021. *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics, Online.
- Loic Barrault, Ondřej Bojar, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. [Findings of the 2019 conference on machine translation \(WMT19\)](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.
- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéal, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. [Findings of the 2016 conference on machine translation](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. [Better hypothesis testing for statistical machine translation: Controlling for optimizer instability](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

- Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. [Transformer-XL: Attentive language models beyond a fixed-length context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tobias Domhan and Felix Hieber. 2017. [Using target-side monolingual data for neural machine translation through multi-task learning](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Marcello Federico, Sebastian Stüker, and François Yvon, editors. 2014. *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*. Lake Tahoe, California.
- Markus Freitag, Matthias Huck, and Hermann Ney. 2014. [Jane: Open source machine translation system combination](#). In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Gothenburg, Sweden. Association for Computational Linguistics.
- Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. [Soft contextual data augmentation for neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.
- Yingbo Gao, Baohao Liao, and Hermann Ney. 2020a. [Unifying input and output smoothing in neural machine translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4361–4372.
- Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020b. [Towards a better understanding of label smoothing in neural machine translation](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223, Suzhou, China. Association for Computational Linguistics.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. [Convolutional sequence to sequence learning](#). In *International conference on machine learning*, pages 1243–1252. PMLR.
- Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. [Generalizing back-translation in neural machine translation](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52, Florence, Italy. Association for Computational Linguistics.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. [Non-autoregressive neural machine translation](#). *arXiv preprint arXiv:1711.02281*.
- Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Hui-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. [On using monolingual corpora in neural machine translation](#). *arXiv preprint arXiv:1503.03535*.
- Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017a. [On integrating a language model into neural machine translation](#). *Computer Speech & Language*, 45:137–148.
- Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017b. [On integrating a language model into neural machine translation](#). *Computer Speech & Language*, 45:137–148.
- Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. 2022. [Language models are general-purpose interfaces](#). *arXiv preprint arXiv:2206.06336*.
- Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, et al. 2022. [General-purpose, long-context autoregressive modeling with perceiver ar](#). *arXiv preprint arXiv:2202.07765*.
- Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. [Layer-wise coordination between encoder and decoder for neural machine translation](#). *Advances in Neural Information Processing Systems*, 31.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. [Learning distributed representations of sentences from unlabelled data](#). In *Proceedings of the 2016 Conference of the North American Chapter of*

- the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.
- Kazuki Irie. 2020. *Advancing Neural Language Modeling in Automatic Speech Recognition*. Ph.D. thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany.
- Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. Language modeling with deep transformers. In *INTERSPEECH*.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709.
- Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. *Effective cross-lingual transfer of neural machine translation models without shared vocabularies*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy. Association for Computational Linguistics.
- Yunsu Kim, Jiahui Geng, and Hermann Ney. 2018. *Improving unsupervised word-by-word translation with language model and denoising autoencoder*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 862–868, Brussels, Belgium. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. *Moses: Open source toolkit for statistical machine translation*. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. *Learning language specific sub-network for multilingual machine translation*. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.
- Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.
- Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239. IEEE.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *Bleu: a method for automatic evaluation of machine translation*. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *Pytorch: An imperative style, high-performance deep learning library*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Ofir Press and Noah A. Smith. 2018. You may not need attention. *ArXiv*, abs/1810.13409.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. *Exploring the limits of transfer learning with a unified text-to-text transformer*. *Journal of Machine Learning Research*, 21(140):1–67.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. *Improving neural machine translation models with monolingual data*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. *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.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. *Self-attention with relative position representations*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- 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.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. [Understanding neural machine translation by simplification: The case of encoder-free models](#). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1186–1193, Varna, Bulgaria. INCOMA Ltd.
- Jörg Tiedemann. 2012. [Parallel data, tools and interfaces in OPUS](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).
- Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Thirty-first AAAI conference on artificial intelligence*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- 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.
- Shuo Wang, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. 2021a. Language models are good translators. *arXiv preprint arXiv:2106.13627*.
- Weiyue Wang, Tamer Alkhouli, Derui Zhu, and Hermann Ney. 2017. Hybrid neural network alignment and lexicon model in direct hmm for statistical machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 125–131.
- Weiyue Wang, Zijian Yang, Yingbo Gao, and Hermann Ney. 2021b. Transformer-based direct hidden markov model for machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 23–32.
- Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan, and Hermann Ney. 2018. [Neural hidden Markov model for machine translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Melbourne, Australia. Association for Computational Linguistics.
- Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.
- Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012. [Jane 2: Open source phrase-based and hierarchical statistical machine translation](#). In *Proceedings of COLING 2012: Demonstration Papers*, pages 483–492, Mumbai, India. The COLING 2012 Organizing Committee.
- Biao Zhang, Behrooz Ghorbani, Ankur Bapna, Yong Cheng, Xavier Garcia, Jonathan Shen, and Orhan Firat. 2022. Examining scaling and transfer of language model architectures for machine translation. *arXiv preprint arXiv:2202.00528*.

Appendix A Grid Search Over Source Reconstruction Settings

architecture	source reconstruction variant				IWSLT14 de-en		WMT16 en-ro		
	task	mask	noise	schedule	BLEU	TER	BLEU	TER	
encoder-decoder	-	-	-	-	34.9	44.5	26.0	54.8	
encoder-only	LM	triangular	none	0	33.5	46.0	25.4	55.5	
				1	34.4	45.3	25.2	55.7	
				lin	34.2	45.2	25.5	55.4	
			exp	34.6	45.2	25.3	55.7		
			BERT	0	33.6	45.7	25.4	55.6	
				1	34.4	45.1	25.2	55.8	
		lin		34.4	45.4	25.6	55.5		
		full	none	0	34.5	44.9	25.8	55.4	
				1	34.5	44.8	25.9	55.0	
				lin	34.5	44.9	25.7	55.3	
			exp	34.4	44.8	26.1	54.8		
			BERT	0	34.5	45.1	26.2	54.8	
				1	34.4	44.9	25.6	55.3	
		lin		34.7	44.5	25.8	55.3		
		AE	triangular	none	0	32.2	47.2	25.3	55.6
					1	32.5	46.2	24.9	55.9
					lin	32.0	46.3	25.2	55.8
				exp	32.0	46.8	25.3	55.3	
	BERT			0	30.8	47.9	25.1	56.1	
				1	33.5	45.9	25.1	56.0	
			lin	31.5	47.5	25.2	55.7		
	full		none	0	34.4	45.1	25.8	55.2	
				1	34.0	45.3	25.9	55.1	
				lin	33.8	45.7	25.7	55.3	
			exp	34.0	45.5	25.7	55.3		
			BERT	0	34.9	45.0	25.8	55.3	
				1	35.0	45.0	26.0	55.1	
	lin			34.7	45.0	25.7	55.4		
	exp		34.8	44.8	25.8	55.4			

Table 9: Grid search of four source-reconstruction-related hyperparameters on de-en and en-ro. LM means to shift the source-side outputs and the auxiliary task corresponds to autoregressive language modeling, and AE means to not shift and corresponds to an autoencoding task. Our interpretations of the table are given in Sec.4.1