One Source, Two Targets: Challenges and Rewards of Dual Decoding

Jitao Xu, François Yvon

▶ **To cite this version:**

Jitao Xu, François Yvon. One Source, Two Targets: Challenges and Rewards of Dual Decoding. Conference on Empirical Methods in Natural Language Processing, Nov 2021, Online and Punta Cana, Dominican Republic. hal-03345478

HAL Id: hal-03345478
https://hal.science/hal-03345478

Submitted on 15 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

One Source, Two Targets: Challenges and Rewards of Dual Decoding

Jitao Xu
Univ. Paris-Saclay,
& CNRS, LISN
Orsay, France
jitao.xu@limsi.fr

François Yvon
Univ. Paris-Saclay,
& CNRS, LISN
Orsay, France
francois.yvon@limsi.fr

Abstract

Machine translation is generally understood as generating one target text from an input source document. In this paper, we consider a stronger requirement: to jointly generate two texts so that each output side effectively depends on the other. As we discuss, such a device serves several practical purposes, from multi-target machine translation to the generation of controlled variations of the target text. We present an analysis of possible implementations of dual decoding, and experiment with four applications. Viewing the problem from multiple angles allows us to better highlight the challenges of dual decoding and to also thoroughly analyze the benefits of generating matched, rather than independent, translations.

1 Introduction

Neural Machine Translation (NMT) is progressing at a rapid pace. Since the introduction of the first encoder-decoder architecture (Sutskever et al., 2014; Cho et al., 2014), then completed with an attention mechanism (Bahdanau et al., 2015; Vaswani et al., 2017), the performance of NMT systems is now good enough for a growing number of services, both for the general public and the translation industry. Not only are neural translation systems better, they are also more versatile and have been extended in many ways to meet new application demands. This is notably the case with multilingual extensions (Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017), which aim to develop systems capable of processing multiple translation directions with one single model.

Another common situation for MT applications is the multi-source / multi-target scenario, where source documents in language S_l need to be published in several target languages $T_l^1, T_l^2, \ldots$. This is, for instance, what happens in multilingual institutions, or with crowdsourced translations of TV shows. The multi-source way (Och and Ney, 2001; Schwartz, 2008; Zoph and Knight, 2016) to handle this generates a first translation into target language T_l^1, which, once revised, can be used in conjunction with the original source to generate the translation into language T_l^2. The expected benefit of this approach is to facilitate word disambiguation.

f	I could do that again if you want .
$\mathbf{e}_1$	只要 你 愿意 我 可以 重复 一遍 。
$\mathbf{e}_2$	もう 一 回 やり ましょ う か
$\mathbf{e}_1$	Je peux le refaire si vous le voulez .
$\mathbf{e}_2$	. voulez le vous si refaire le peux Je
$\mathbf{e}_1$	Ich kann das noch mal machen , wenn Sie wollen .
$\mathbf{e}_2$	Ich kann das noch mal machen , wenn du willst .

Table 1: Instances of Dual Decoding: multi-target translation (§ 3), bi-directional decoding (§ 4), variant generation (§ 6).

An alternative, that we thoroughly explore here, is to *simultaneously generate* translations in T_l^1 and T_l^2, an approach termed *multi-target translation* by Neubig et al. (2015). While the same goal is achieved with a multilingual system translating independently in T_l^1 and T_l^2, several pay-offs are expected from a joint decoding: (a) improved disambiguation capacities (as for multi-source systems); (b) a better collaboration between the stronger and the weaker decoders; (c) more consistent translations in T_l^1 and T_l^2 than if they were performed independently. As it turns out, a dual decoder computing joint translations can be used for several other purposes, that we also consider: to simultaneously *decode in two directions*, providing a new implementation of the idea of Watanabe and Sumita (2002); Finch and Sumita (2009); to *disentangle mixed language* (code-switched) texts into their two languages (Xu and Yvon, 2021); finally, to generate *coherent translation alternatives*, an idea we use to compute polite and impolite variants of the same input (Sennrich et al., 2016a). Considering multiple applications allows us to assess the challenges and rewards of dual decoding under various angles and to better evaluate the actual

agreement between the two decoders' outputs. Our main contributions are the following: (i) a comparative study of architectures for dual decoding (§ 2); (ii) four short experimental studies where we use these architectures to simultaneously generate several outputs from one input (§ 3–§ 6); (iii) practical remedies to the shortage of multi-parallel corpora that are necessary to implement multi-target decoding; (iv) concrete solutions to mitigate exposure bias between two decoders; (v) quantitative evaluations of the increased consistency incurred by a tight interaction between decoders. An additional empirical finding that is of practical value is the benefits of exploiting multi-parallel corpora to finetune multilingual systems.

2 Architectures for Dual Decoding

2.1 Model and Notations

In our setting, we consider the simultaneous translation of sentence $\mathbf{f}$ in source language S_l into two target sentences $\mathbf{e}^1$ and $\mathbf{e}^2$ in languages[1] T_l^1 and T_l^2. In this situation, various modeling choices can be entertained (Le et al., 2020):

$$P(\mathbf{e}^1, \mathbf{e}^2|\mathbf{f}) = \prod_{t=1}^{T} P(\mathbf{e}_t^1, \mathbf{e}_t^2|\mathbf{f}, \mathbf{e}_{<t}^1, \mathbf{e}_{<t}^2) \quad (1)$$

$$P(\mathbf{e}^1, \mathbf{e}^2|\mathbf{f}) = \prod_{t=1}^{T} P(\mathbf{e}_t^1|\mathbf{f}, \mathbf{e}_{<t}^1, \mathbf{e}_{<t}^2) \times P(\mathbf{e}_t^2|\mathbf{f}, \mathbf{e}_{<t}^1, \mathbf{e}_{<t}^2) \quad (2)$$

$$P(\mathbf{e}^1, \mathbf{e}^2|\mathbf{f}) = \prod_{t=1}^{T} P(\mathbf{e}_t^1|\mathbf{f}, \mathbf{e}_{<t}^1) P(\mathbf{e}_t^2|\mathbf{f}, \mathbf{e}_{<t}^2), \quad (3)$$

where $T = max(|\mathbf{e}^1|, |\mathbf{e}^2|)$, and we use placeholders whenever necessary. The factorization in Equation (1) implies a joint event space for the two languages and a computational cost we deemed unreasonable. We instead resorted to the second (*dual*) formulation, that we contrasted with the third one (*independent* generation) in our experiments. Note that thanks to asynchronous decoding, introduced in Section 2.4.2, we are also in a position to simulate other dependency patterns, where each symbol $\mathbf{e}_t^2$ is generated conditioned on $\mathbf{e}_{<t+k}^1, \mathbf{e}_{<t}^2$, thus reproducing the *chained* model of Le et al. (2020).

[1] In our applications, these do not always correspond to actual natural languages. We keep the term for expository purposes.

2.2 Attention Mechanism

Our dual decoder model implements the encoder-decoder architecture of the Transformer model of (Vaswani et al., 2017). In this model, the input to each attention head consists of queries $\mathbf{Q}$, key-value pairs $\mathbf{K}$ and $\mathbf{V}$. Each head maps a query and a set of key-value pairs to an output, computed as a weighted sum of the values, where weights are based on a compatibility assessment between query and keys, according to (in matrix notations):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \quad (4)$$

with d_k the shared dimension of queries and keys. Note that $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ for each head are obtained by linearly transforming the hidden states from previous layer with different projection matrices.

2.3 Proposal for a Dual Decoder

We chose to implement Equation (2) with a synchronous coupling of two decoders sharing the same encoder. An alternative would be to have the two decoders share the bottom layers and only specialize at the upper layer(s) for one specific language: we did not explore this idea further, as it seemed less appropriate for the variety of applications considered. Figure 1 illustrates this design. Compared to a standard Transformer, we add a cross attention layer in each decoder block to capture the interaction between the two decoders. Denoting the output hidden states of the previous layer for each decoder as H_l^1 and H_l^2, the decoder cross-attention is computed as:[2]

$$\begin{aligned} H_{l+1}^1 &= \text{Attention}(H_l^1, H_l^2, H_l^2) \\ H_{l+1}^2 &= \text{Attention}(H_l^2, H_l^1, H_l^1), \end{aligned} \quad (5)$$

where Attention is defined in Equation (4). The two decoders are thus fully synchronous as each requires the hidden states of the other in each block to compute its own hidden states. The decoder cross-attention can be inserted before or after the encoder-decoder attention. Preliminary experiments with these variants have shown that they were performing similarly. We thus only report results obtained with the decoder cross-attention as the last attention component of a block (see Figure 1).

[2] For simplicity, we omit the other sub-layers (self attention, encoder-decoder cross attention, feed forward and layer normalization).

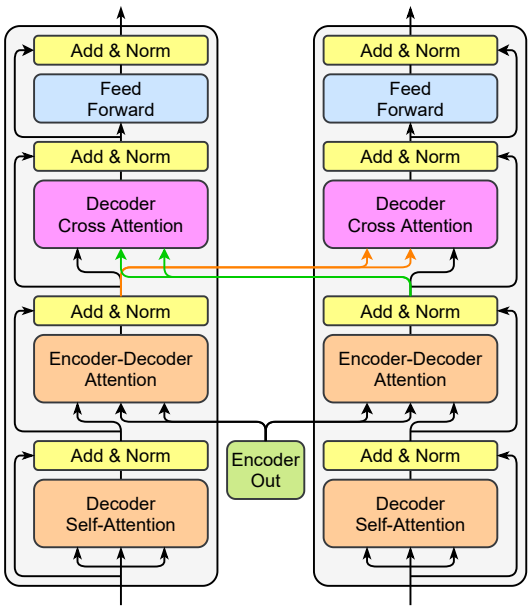

Figure 1: A graphical view of the dual decoder.

2.4 Synchronous Beam Search

2.4.1 Full Synchronous Mode

Our decoding algorithm uses a dual beam search. Assuming each decoder uses its own beam of size k, the cross-attention between decoders can be designed and implemented in multiple ways: for instance, one could have each hypothesis in decoder 1 attend to any hypotheses in decoder 2, which would however create an exponential blow-up of the search space. Following Zhou et al. (2019), we only compute the attention between 1-best candidates of each decoder, 2-best candidates in each decoder, etc. This heuristic ensures that the number of candidates in each decoder beam remains fixed. There is however an added complexity, due to the fact that the ranking of hypotheses in each decoder beam evolves over time: the best hypothesis in decoder 1 at time t may no longer be the best at time $t+1$. Preserving the consistency of the decoder states therefore implies to recompute the entire prefix representation for each hypothesis and each decoder at each time step, thus creating a significant computing overhead.[3]

We also explored other implementations where each candidate prefix in one beam always attends to the best candidate in the other beam or attends to the average of all candidates. These variants ended up delivering very similar results, as also found in (Zhou et al., 2019). For simplicity reasons, we use the first scheme in all experiments.

[3]In our experiments, the dual decoder was about twice as slow as the two independent ones.

2.4.2 Relaxing Synchronicity

Simultaneously generating symbols in two languages is a very strong requirement, and may not bring out all the benefits of dual decoding, especially when the two target languages have different word orders. We relax this assumption by allowing one decoder to start generating symbols before the other: this is implemented by having the delayed decoder generate dummy symbols for a fixed number of steps before generating meaningful words, a strategy akin to the wait-k approach in spoken language translation (Elbayad et al., 2020).

A more extreme case of delayed processing is when one decoder can access a *complete translation* in the other language. In our implementation, this is simulated with partial forced decoding, where one translation is predefined, while the other is computed. We explored this in two settings: (a) within a *two-pass, sequential procedure*, where the output of step 1 for decoder 1 is fully known and fixed when computing the second output of decoder 2; (b) using a *reference translation* in one of the decoder, implementing a controlled decoding where the output in T_l^2 not only translates the source, but does so in a way that is consistent with the reference translation in T_l^1. These strategies are used in Sections 3 and 6.

2.5 Training and Fine-tuning

Training this model requires triplets of instances comprising one source and two targets. Given a set of such examples $D = \{(\mathbf{f}, \mathbf{e}^1, \mathbf{e}^2)_i, i = 1 \dots N\}$, the training maximizes the combined log-likelihood for the two target sequences:

$$L(\theta) = \sum_D \Big(\sum_{t=1}^{|\mathbf{e}^1|} \log P(\mathbf{e}_t^1 | \mathbf{e}_{<t}^1, \mathbf{e}_{<t}^2, \mathbf{f}, \theta) + \sum_{t=1}^{|\mathbf{e}^2|} \log P(\mathbf{e}_t^2 | \mathbf{e}_{<t}^2, \mathbf{e}_{<t}^1, \mathbf{f}, \theta)\Big), \quad (6)$$

where θ represents the set of parameters.

As multi-parallel corpora are not as common as bilingual ones, we also considered a two-step procedure which combines bilingual and trilingual data. In a first step, we train a standard multilingual model (one monolingual encoder, one bilingual decoder), where tags are used to select the target language (Johnson et al., 2017). This only requires bilingual data $\{(\mathbf{f}, \mathbf{e}^1)_i, i = 1 \dots N^1\}$ and

$\{(\mathbf{f}', \mathbf{e}^2)_j, j = 1 \dots N^2\}$. We then initialize the dual decoder model with pre-trained parameters and fine-tune with the trilingual dataset. Both decoders thus start with the same pre-trained decoder. The decoder cross-attention matrices cannot benefit from pre-training and are initialized randomly. During fine-tuning, tags are no longer necessary as both target translations are required.

3 Multi-target Machine Translation

3.1 Data

We first evaluate our dual decoder model on the multi-target MT task for three directions: English to German/French (En→De/Fr), German to English/French (De→En/Fr) and English to Chinese/Japanese (En→Zh/Ja). Similarly to (Wang et al., 2019; He et al., 2021), we use the IWSLT17 dataset[4] (Cettolo et al., 2012) as our main test bed.[5] Pre-training experiments additionally use WMT20 De-En, De-Fr, En-Zh, En-Ja and WMT14 En-Fr bilingual datasets.[6] We use the IWSLT `tst2012` and `tst2013` as development sets and test our model on `tst2014`.[7] Table 2 summarizes the main statistics for trilingual training and test data.

	Original De	Original Fr	3-way
Train	209522	236653	205397
Dev	2693	3083	2468
tst2014	1305	1306	1168
	Original Zh	Original Ja	3-way
Train	235078	226834	213090
Dev	3064	3024	2837
tst2014	1297	1285	1214

Table 2: Number of lines in the trilingual IWSLT data. English is used to identify trilingual sentences and is therefore not shown in this table.

For WMT data, we discard sentence pairs with invalid language tag as computed by `fasttext` language identification model[8] (Bojanowski et al., 2017). We tokenize all English, German and French data using Moses tokenizer.[9] Chinese and Japanese sentences are segmented using `jieba`[10] and `mecab`[11] respectively. For En→De/Fr and De→En/Fr, we use a shared source-target vocabulary built with 40K Byte Pair Encoding (BPE) units (Sennrich et al., 2016b) learned on WMT data with `subword-nmt`.[12] For En→Zh/Ja, we build a 32K BPE model for En and a joint 32K BPE for Zh and Ja, both learned on the WMT data.[13]

3.2 Experimental Settings

We implement the dual decoder model using `fairseq`[14] (Ott et al., 2019),[15] with a hidden size of 512 and a feedforward size of 2048. We optimize with Adam, set up with a maximum learning rate of 0.0007 and an inverse square root decay schedule, as well as 4000 warmup steps. For fine-tuning models, we use Adam with a fixed learning rate of 8e−5. For standard Transformer models, we share the decoder input and output matrices, while for dual decoder models, we share all four input and output decoder matrices (Press and Wolf, 2017; Inan et al., 2017). All models are trained with mixed precision and a batch size of 8192 tokens on 4 V100 GPUs. Pre-training last for 300k iterations, while all other models are trained until no improvement is found for 4 consecutive checkpoints on the development set. Performance is computed with SacreBLEU (Post, 2018).

We call the dual decoder models `dual`. To study the effectiveness of dual decoding, we also train a simplified multi-task model (Dong et al., 2015), implementing the independent model of Equation (3) without decoder cross-attention. For `indep`, the only interaction between outputs is thus a shared loss computed on multi-parallel data. Baseline Transformer models trained separately on each language pair are denoted by `base`.

3.3 Results

We evaluate the performance of models trained only with trilingual data, as well as models pre-trained in a multilingual way. Table 3 shows that the `indep` model outperforms the `base` model in all directions, demonstrating the benefits of jointly training two independent decoders. The same gain is not observed for the `dual` model, for which results in some directions are even worse than the baseline.

[4] https://wit3.fbk.eu/2017-01-c

[5] Subject to some filtering to obtain a fully multi-parallel set of sentences.

[6] See http://statmt.org.

[7] For comparison with (He et al., 2021), we also report the results on `tst2015` in Appendix D.

[8] https://dl.fbaipublicfiles.com/fasttext/supervised-models/lid.176.bin

[9] https://github.com/moses-smt/mosesdecoder

[10] https://github.com/fxsjy/jieba

[11] https://taku910.github.io/mecab/

[12] https://github.com/rsennrich/subword-nmt

[13] Full experimental details are in Appendix A.

[14] https://github.com/pytorch/fairseq

[15] Our implementation is open-sourced at https://github.com/jitao-xu/dual-decoding.

Model	De-En	De-Fr	SIM	En-De	En-Fr	SIM	En-Zh	En-Ja	SIM	Avg BLEU	Avg SIM
`base`	32.6	24.8	89.28	28.1	38.8	91.34	22.2	13.6	81.97	25.5	87.53
`multi`	31.1	24.4	91.56	25.9	37.9	91.87	25.3	10.4	83.71	25.8 (+0.3)	89.05 (+1.52)
`indep`	33.8	25.4	90.04	29.1	39.8	91.63	22.6	14.8	83.16	26.3 (+0.8)	88.28 (+0.75)
`dual`	31.8	22.4	90.60	28.5	38.8	91.45	22.8	15.3	84.09	25.6 (+0.1)	88.71 (+1.18)
`indep` ps	33.4	26.1	90.51	28.5	39.6	91.98	22.7	14.3	83.58	26.2 (+0.7)	88.69 (+1.16)
`dual` ps	33.2	25.9	91.01	28.4	39.7	92.07	22.5	14.3	83.92	26.2 (+0.7)	89.00 (+1.47)
`indep` FT	37.1	28.6	91.52	30.1	42.3	92.25	26.5	17.1	84.86	30.3 (+4.8)	89.54 (+2.01)
`dual` FT	37.1	28.0	91.53	30.0	42.3	92.43	26.0	17.0	85.03	30.1 (+4.6)	89.66 (+2.13)
`indep` FT+ps	36.5	28.2	92.17	29.9	42.0	92.68	25.9	16.3	84.76	29.8 (+4.3)	89.87 (+2.34)
`dual` FT+ps	36.5	28.4	92.26	30.1	42.1	92.55	26.1	16.5	84.84	30.0 (+4.5)	89.88 (+2.35)

Table 3: BLEU and similarity scores of multi-target models. Similarity scores (SIM) are computed as the cross-lingual similarity between the two target translations. Pseudo (ps) refers to models trained from scratch with synthetic reference data. FT indicates models fine-tuned from the pre-trained multilingual (`multi`) model. FT+ps refers to models fine-tuned using synthetic reference data.

One explanation is that dual decoding suffers from a double exposure bias, as errors in both decoders jointly contribute to derail the decoding process. We get back to this issue in Section 4.

To test this, we use the `base` model to translate source texts $\mathbf{f}$ into targets $\hat{\mathbf{e}}^1$ and $\hat{\mathbf{e}}^2$, which are then merged with the original data to build a pseudo-trilingual training set. For fair comparison, we only use half of the translations from each target language, yielding a pseudo-trilingual dataset $\{(\mathbf{f}_{1/2}, \hat{\mathbf{e}}^1_{1/2}, \mathbf{e}^2_{1/2}), (\mathbf{f}_{2/2}, \mathbf{e}^1_{2/2}, \hat{\mathbf{e}}^2_{2/2})\}$ that is as large as the original data. We see in Table 3 that these artificial translations almost close the gap between the independent and dual decoders.

Initializing with pre-trained models (Section 2.5) brings an additional improvement for both methods, thus validating our pre-training procedure (see bottom of Table 3). They confirm that dual decoders can be effectively trained, even in the absence of large multi-parallel data. These results also highlight the large gains of fine-tuning on a tri-parallel corpus, which improves our baseline multilingual models by nearly 5 BLEU points on average.

We additionally experiment fine-tuning pre-trained models with the synthetic pseudo-trilingual data. This setting (FT+ps in Table 3) does not bring any gain in translation quality: for the `indep` model we see a small loss due to training with noisy references; for `dual`, it seems that mitigating exposure bias is less impactful when starting from well-trained models.

3.4 Complements and Analysis

The value of dual decoding is to ensure that translations $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are more consistent than with independent decoding. To evaluate this, we compute the similarity scores (SIM) between these two translations using LASER.[16] As shown in Table 3, `dual` model generate translations that are slightly more similar on average than the `indep` model: as both translate the same source into the same languages, similarity scores are always quite high.

Model	En-De	En-Fr
`dual`	28.5	38.8
`dual` De+Fr auto	28.7	39.3
`dual` De+Fr ref	28.7	39.6
`dual` FT	30.0	42.3
`dual` FT De wait-3	30.2	42.4
`dual` FT Fr wait-3	30.4	42.6
`dual` FT De+Fr auto	30.0	42.5
`dual` FT De+Fr ref	30.0	42.4

Table 4: BLEU scores for asynchronous decoding: sequential decoding on the `dual` model trained from scratch (top), wait-k models fine-tuned on the pre-trained model and sequential decoding on the `dual` FT model (bottom) for the direction En→De/Fr. Results using sequential decoding for one decoder are obtained in a second decoding pass using either automatic (auto) or reference (ref) translations.

As explained in Section 2.4, the dual decoder model is not limited to strictly synchronous generation and accommodates relaxed variants (as well as alternative dependency patterns) where one decoder can start several steps after the other. We fine-tune "wait-k" `dual` models from the pre-trained model with $k = 3$ for En→De/Fr and evaluate the effects on performance. As shown in Table 4, the BLEU scores are slightly improved for both targets when either side is delayed by 3 steps. These results suggest that depending on language pairs, the information flow between decoders can be benefi-

[16] https://github.com/facebookresearch/LASER

Model	En-De	Cons	En-Fr	Cons	En-Zh	Cons	En-Ja	Cons	Avg BLEU	Avg Cons
base	28.1	-	38.8	-	22.2	-	13.6	-	25.7	-
indep	29.1	54.7	39.4	65.8	22.5	51.3	14.8	37.6	26.5 (+0.8)	52.4
dual	25.9	88.9	36.6	90.5	20.6	86.0	4.2	68.4	21.8 (-3.9)	83.5 (+31.1)
indep pseudo	29.0	65.7	39.9	73.3	22.9	62.0	15.6	48.6	26.9 (+1.2)	62.4
dual pseudo	28.7	83.7	38.9	89.6	23.1	80.2	15.1	67.5	26.5 (+0.8)	80.3 (+17.9)
indep pseudo-dup	29.3	70.9	40.5	76.3	23.5	67.6	15.8	53.7	27.3 (+1.6)	67.1
dual pseudo-dup	29.6	83.5	40.1	89.6	23.4	78.2	15.3	70.7	27.1 (+1.4)	80.5 (+13.4)

Table 5: Results of bi-directional MT models trained with actual data (top) and synthetic data (bottom). The consistency score (Cons) is an averaged BLEU score between the forward and backward translations.

cial from a small amount of asynchronicity.

Our implementation also enables to have one decoder finish before the other begins. We thus experiment a *sequential decoding strategy* (see Section 2.4.2), in which we first compute the complete translation in one target language (with the dual model), then decode the other one. In this case, the second decoding step has access to both the source and the other target sequence. This decoding strategy does not require any additional training and is applied directly during inference.

We decode both dual and dual FT models with this strategy. Results in Table 4, obtained with both automatic and reference translations in one language, show that this technique is able to improve the dual model on both French and German translations, while only slightly improves the French translation for the dual FT model. Sequential decoding with reference in one language provides the other decoder with the ground truth, which therefore alleviates the exposure bias problem suffered by dual models. However, combining results of FT models in Table 3 and 4, we see that fine-tuned models are less sensitive to errors made during decoding. This again shows the benefit that dual models actually obtain from pretrained models.

4 Bi-directional MT

Bi-directional MT (Finch and Sumita, 2009; Zhou et al., 2019; Liu et al., 2020b) aims to integrate future information in the decoder by jointly translating in the forward (left to right, L2R) and in the backward (right to left, R2L) directions. Another expectation is that the two decoders, having different views of the source, will deliver complementary translations. Dual decoding readily applies in this setting, with one decoder for each direction, with the added benefit of generating more coherent outputs than independent decoders. We evaluate this added consistency by reusing the experimental setting (data, implementation and hyperparameters) of Section 3, and by training 4 bi-directional systems, from English into German, French, Chinese and Japanese. Similar to Zhou et al. (2019), we output the translation with the highest probability, inverting the translation if the R2L output is picked.

We first train models on tri-parallel corpora obtained by adding an inverted version of the target sentence to each training sample. In this setting, the dual model again suffers a clear drop of BLEU scores as compared to indep model (Table 5). We again attribute this loss to the impact of the exposure bias, as can be seen in Table 5, where the loss in BLEU score of the dual system is accompanied by a very large increase in consistency of the outputs (+31.1). We therefore again introduce pseudo-parallel targets, where one of the two targets is automatically generated with the base model. This was also proposed in (Zhou et al., 2019; Wang et al., 2019; Zhang et al., 2020; He et al., 2021). Similar to the pseudo-data described in Section 3.3, we generate a pseudo dataset in which each original source sentence occurs just once. This means that the forward and backward training target sentences are not always deterministically related, which forces each decoder to put less trust on tokens from the other direction. We also consider the pseudo-dup data, in which each source sentence is duplicated, occurring once with the reference data in each direction. Results in Table 5 show that this method again closes the gap between indep and dual, and yields systems that surpass the baseline by about 1 BLEU point in the pseudo setting, and by 1.5 BLEU point in the pseudo-dup setting.

By computing the BLEU score between the two output translations, we can also evaluate the increment of consistency incurred in dual decoding. These scores are reported in Table 5 (column *Cons*)

and show to a +13.4 BLEU increment when averaged over language pairs, thereby demonstrating the positive impact of dual decoding.

5 MT for Code-switched Inputs

In this section, we turn to a novel task, consisting in translating a code-switched (CSW) sentence (containing fragments from two languages) simultaneously into its two components. An example in Table 6 for French-English borrowed from (Carpuat, 2014) illustrates this task.

$\mathbf{f}$:	autrement dit, they are getting out of the closet
$\mathbf{e}^1$:	In other words, they are getting out of the closet
$\mathbf{e}^2$:	autrement dit, ils sortent du placard

Table 6: Dual decoding for a CSW sentence.

Code-switching is an important phenomenon in informal communications between bilingual speakers. It generally consists of short inserts of a secondary language which are embedded within larger fragments in the primary language. When simultaneously translating into these two languages, we expect the following "copy" constraint to be satisfied: *every word in the source text should appear in at least one of the two outputs*.

Our main interest in this experiment is to assess how much dual decoding actually enforces this constraint. As tri-parallel corpora for this task are scarce (Menacer et al., 2019), we mostly follow (Song et al., 2019; Xu and Yvon, 2021) and automatically generate artificial CSW sentences from regular parallel data. Working with the En-Fr pair, we use the WMT14 En-Fr data to generate training data as well as a CSW version of the `newstest2014` test set. Approximately half of the test sentences are mostly English with inserts in French, and mostly French with inserts in English for the other half. We use the same pretraining procedure as in Section 3.2 and evaluate with `csw-newstest2014` data.

Table 7 reports overall BLEU scores, as well as scores for the 'primary and 'secondary' part of the test set for each target language. These results show that `indep` and `dual` systems, which are both able to translate French mixed with English and English mixed with French, achieve performance that is comparable to the `base` model, which, in this experiment, *is made of two distinct Transformer models*, one for each direction.

We also measure how well the constraint expressed above is satisfied. It stipulates that every token in a CSW sentence should be either copied in one language (and translated into the other), or copied in both, which mostly happens for punctuations, numbers or proper names. Our analysis in Table 8 shows that the `base` model is more likely to reproduce the patterns observed in the reference, notably is less likely to generate two copies for a token than the other systems. However, `indep` and, to a larger extend, `dual`, are able to reduce the rate of *lost tokens*, i.e. of source tokens that are not found in any output. This again shows that the interaction between the two decoders helps to increase the consistency between the two outputs.

Model	CSW-En		CSW-Fr	
	second	primary	second	primary
`base`	67.8		67.4	
	35.5	97.4	37.5	95.3
`indep`	67.7		67.3	
	35.1	97.4	37.1	95.5
`dual`	67.7		67.5	
	35.1	97.5	37.5	95.4

Table 7: BLEU scores of CSW translation models tested on the `csw-newstest2014` data that we generated. Small numbers are scores computed separately on the two parts of the test set where the target language is primary or secondary (second).

Model	Exclusive	Both	Punctuations	Lost
reference	81.56	8.10	10.34	0
`base`	79.14	8.85	11.29	0.72
`indep`	78.86	9.13	11.35	0.67
`dual`	78.90	9.17	11.32	0.61

Table 8: Analysis of the "copy" constraint. "Exclusive" refers to the percentage of test tokens appearing in only one hypothesis. "Both" and "Punctuations" are for tokens and punctuations+digits appearing in both hypotheses, and "Lost" is for tokens not found in either.

6 Generating Translation Variants

As a last application of dual decoding, we study the generation of pairs of consistent translation alternatives, using variation in "politeness" as our test bed. We borrow the experimental setting and data of Sennrich et al. (2016a).[17] The training set contains 5.58M sentences pairs, out of which 0.48M are annotated as polite and 1.06M as impolite. The rest is deemed neutral.[18] Using this data, we generate tri-parallel data as follows. We first train a tag-based

[17] `http://data.statmt.org/rsennrich/politeness/`

[18] See details in Appendix C.

NMT with politeness control as in Sennrich et al. (2016a) and use it to predict the polite counterpart of each impolite sentence, and vice-versa. We also include an equivalent number of randomly chosen neutral sentences: for these, the polite and impolite versions are identical. The resulting 3-way corpus contains 3.07M sentences. Similar to the multi-target task (Section 3), we fine-tune a pre-trained model with this data until convergence. We use the `test` data of Sennrich et al. (2016a) as development set and test our model on the `testyou` set, which contains 2k sentences with a second-person pronoun *you(r(s(elf)))* in the English source. The annotation tool distributed with the data is used to assess the politeness of the output translations.

Table 9 (top) reports the performance of the pre-trained model. *ref* refers to the annotation result of the reference German sentences. *none* is translated without adding any tags to the source text, while *pol* and *imp* are translated with all sentences tagged respectively as *polite* and *impolite*. The *oracle* line is obtained by prefixing each source sentence with the correct tag. These results show the effectiveness of side constraints for the generation of variants: for both polite and impolite categories, the pre-trained model generates translations that mostly satisfy the desired requirement.

Model	Tag	neutral	pol	imp	BLEU
	ref	438	525	1037	-
`pre-train`	none	1914	16	70	17.7
	pol	479	1518	3	20.9
	imp	22	0	1978	24.1
	oracle	551	406	1043	30.2
	Dec	neutral	pol	imp	BLEU
`indep`	pol	528	1470	2	21.0
	imp	82	0	1918	24.4
`dual`	pol	541	1457	2	21.3
	imp	97	0	1903	24.3
seq imp	pol	531	1467	2	21.2
seq pol	imp	84	0	1916	24.4

Table 9: Results of politeness MT models. Tags are used for the `pre-train` model to generate the desired variant. Decoders (Dec) of `indep` and `dual` compute two translations in one decoding step, while the results using sequential decoding for one decoder are obtained with the 2-step procedure of Section 2.4.2.

Results of the fine-tuned dual decoder models are in Table 9 (bottom): we see that both models are very close and generate more neutral translations and also slightly improve the BLEU scores compared to the pre-trained model.

As discussed in Section 2.4.2, our dual decoder model can delay one decoder until the other is finished. We redo the same sequential decoding procedure as in Section 3.4. Results in Table 9 (bottom) indicate that given the full translation of impolite variations, the `dual` model tends to generate less neutral sentences but more polite ones. The same phenomenon is also observed in the other direction. This implies that the output variations can be better controlled with sequential decoding.

7 Related Work

The variety of applications considered here makes it difficult to give a thorough analysis of all the related work, and we only mention the most significant landmarks.

Multi-source / Multi-target Machine Translation Multi-source MT was studied in the framework of SMT, considering with a tight integration (in the decoder), or a late integration (by combining multiple hypotheses obtained with different sources). This idea was revisited in the Neural framework (Zoph and Knight, 2016; Liu et al., 2020a). Setting multilingual MT aside (Dabre et al., 2020), studies of the multi-target case are comparatively rarer (Neubig et al., 2015). Notable references are (Dong et al., 2015), which introduces a multi-task framework; (Wang et al., 2018), which studies ways to strengthen a basic multilingual decoder; while closer to our work, Wang et al. (2019) consider a dual decoder relying on dual self-attention mechanism. Related techniques have also been used to simultaneously generate a transcript and a translation for a spoken input (Anastasopoulos and Chiang, 2018; Le et al., 2020) and to generate consistent caption and subtitle for an audio source (Karakanta et al., 2021).

Bi-directional Decoding is an old idea from the statistical MT era (Watanabe and Sumita, 2002; Finch and Sumita, 2009). Instantiations of these techniques for NMT are in (Zhang et al., 2018; Su et al., 2019), where asynchronous search techniques are considered; and in (Zhou et al., 2019; Wang et al., 2019; Zhang et al., 2020) where, similar to our work, various ways to enforce a tighter interaction between directions are considered in synchronous search, while Liu et al. (2020b) also study ways to increase the agreement between L2R and R2L directions. More recently, (He et al., 2021) combines multi-target and bi-directional decoding

within a single architecture, where, in each layer and block, all cross-attentions are combined with a single hidden state; four softmax layers are used for the output symbols in a proposal that creates an even stronger dependency between decoders than what we consider here.

Code-switching is an important linguistic phenomenon in bilingual communities that is getting momentum within the natural language processing communities (Sitaram et al., 2019). Several tasks have been considered: token-level language identification (Samih et al., 2016), Language Modeling (Winata et al., 2019), Named Entity Recognition (Aguilar et al., 2018), Part-of-Speech tagging (Ball and Garrette, 2018) and Sentiment Analysis (Patwa et al., 2020). Machine Translation for CSW texts is considered in (Menacer et al., 2019).

8 Conclusion and Future Work

In this paper, we have explored various possible implementations of dual decoding, as a way to generate pairs of consistent translation. Dual decoding can be viewed as a tight form of multi-task learning, and, as we have seen, can be effectively trained using actual or partly artificial data; it can also directly benefit from pre-trained models. Considering four applications of MT, we have observed that dual decoding was prone to exposure bias in the two decoders, and we have proposed practical remedies. Using these, we have achieved BLEU scores that match those of a simple multi-task learners, and display an increased level of consistency.

In our future work, we plan to consider other strategies, such as scheduled sampling (Bengio et al., 2015; Mihaylova and Martins, 2019), to mitigate the exposure bias. Another area where we seek to improve is the relaxation of strict synchronicity in decoding. We finally wish to study more applications of this technique, notably to generate controlled variation: controlling gender variation (Zmigrod et al., 2019) or more complex form of formality levels, as in (Niu and Carpuat, 2020), are obvious candidates.

Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2021-[AD011011580R1] made by GENCI. The authors wish to thank Josep Crego for his comments and discussions. We would also like to thank the anonymous reviewers for their valuable suggestions. The first author is partly funded by Systran and by a grant from Région Ile-de-France.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Named entity recognition on code-switched data: Overview of the CALCS 2018 shared task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147, Melbourne, Australia. Association for Computational Linguistics.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. 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 82–91, New Orleans, Louisiana. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kelsey Ball and Dan Garrette. 2018. Part-of-speech tagging for code-switched, transliterated texts without explicit language identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3084–3089, Brussels, Belgium. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marine Carpuat. 2014. Mixed language and code-switching in the Canadian hansard. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 107–115, Doha, Qatar. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties

of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Josep M. Crego, José B. Mariño, and Adrià De Gispert. 2005. Reordered search, and tuple unfolding for Ngram-based SMT. In *In Proceedings of the MT Summit X*, pages 283–289.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

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

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Interspeech 2020 - Conference of the International Speech Communication Association*, pages 1461–1465, Shangai (Virtual Conf), China.

Andrew Finch and Eiichiro Sumita. 2009. Bidirectional phrase-based statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1124–1132, Singapore. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875. Association for Computational Linguistics.

Thanh-He Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation*, IWSLT 2016, Vancouver, Canada.

Hao He, Qian Wang, Zhipeng Yu, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2021. Synchronous interactive decoding for multilingual neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12981–12988.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Zhifeng Wu, Yonghui andhen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Alina Karakanta, Marco Gaido, Matteo Negri, and Marco Turchi. 2021. Between flexibility and consistency: Joint generation of captions and subtitles. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 215–225, Bangkok, Thailand (online). Association for Computational Linguistics.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jianfeng Liu, Ling Luo, Xiang Ao, Yan Song, Haoran Xu, and Jian Ye. 2020a. Meet changes with constancy: Learning invariance in multi-source translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1122–1132, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2020b. Agreement on target-bidirectional recurrent neural networks for sequence-to-sequence learning. *Journal of Artificial Intelligence Research*, 67:581–606.

Mohamed Menacer, David Langlois, Denis Jouvet, Dominique Fohr, Odile Mella, and Kamel Smaïli. 2019. Machine Translation on a parallel Code-Switched Corpus. In *Canadian AI 2019 - 32nd Conference on Canadian Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Ontario, Canada.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

Graham Neubig, Philip Arthur, and Kevin Duh. 2015. Multi-target machine translation with multi-synchronous context-free grammars. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 293–302, Denver, Colorado. Association for Computational Linguistics.

Xing Niu and Marine Carpuat. 2020. Controlling neural machine translation formality with synthetic supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8568–8575.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of MT Summit*, Santiago de Compostela, Spain.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via LSTM recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas. Association for Computational Linguistics.

Lane Schwartz. 2008. Multi-source translation methods. In *MT at work: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 279–288, Waikiki, Hawaii.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. 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.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified 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 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinsong Su, Xiangwen Zhang, Qian Lin, Yue Qin, Junfeng Yao, and Yang Liu. 2019. Exploiting reverse target-side contexts for neural machine translation via asynchronous bidirectional decoding. *Artificial Intelligence*, 277:103168.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Yining Wang, Jiajun Zhang, Long Zhou, Yuchen Liu, and Chengqing Zong. 2019. Synchronously generating two languages with interactive decoding. 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 3350–3355, Hong Kong, China. Association for Computational Linguistics.

Taro Watanabe and Eiichiro Sumita. 2002. Bidirectional decoding for statistical machine translation. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Jitao Xu and François Yvon. 2021. Can you traducir this? Machine translation for code-switched input. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 84–94, Online. Association for Computational Linguistics.

Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2020. Synchronous bidirectional inference for neural sequence generation. *Artificial Intelligence*, 281:103234.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5698–5705. AAAI Press.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics*, 7:91–105.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

A Details of Data for Multi-target and Bi-directional Machine Translation

We use the IWSLT17 dataset as training data. We use IWSLT17.TED.tst2012 and IWSLT17.TED.tst2013 as development set and test our model on IWSLT17.TED.tst2014 and IWSLT17.TED.tst2015. The original data is not entirely multi-parallel. Therefore, we extract the shared English sentences from En-De and En-Fr data with the corresponding translation to build a truly trilingual corpus. The En→Zh/Ja trilingual data is built similarly. Table 10 summarizes the statistics for the trilingual training and test data.

	Original De	Original Fr	3-way
Train	209522	236653	205397
Dev	2693	3083	2468
tst2014	1305	1306	1168
tst2015	1080	1210	1074
	Original Zh	Original Ja	3-way
Train	235078	226834	213090
Dev	3064	3024	2837
tst2014	1297	1285	1214
tst2015	1205	1194	1132

Table 10: Statistics of extracted trilingual IWSLT data. English is used to extract trilingual sentences therefore not shown in this table.

We use WMT20 De-En, De-Fr, En-Zh, En-Ja and WMT14 En-Fr bilingual data for our pre-training experiments. For De-En, De-Fr and En-Fr, we discard the ParaCrawl data and use all the rest. For En-Zh, we only use News Commentary, Wiki Titles, CCMT corpus and WikiMatrix data. For En-Ja, we use all data except ParaCrawl and TED talks. The latter is our trilingual data that we do not use in our pre-training stage. For all WMT data, we discard sentence pairs with invalid language tag as computed by `fasttext` language identification model [19] (Bojanowski et al., 2017). Detailed statistics for the WMT data that we have actually used for each language pair are in Table 11.

To generate the pseudo data, taking En→De/Fr as an example, we first train individual Transformer models for En→De and En→Fr using the trilingual data. We then use the En→De model to translate half of the English source $\mathbf{f}_{1/2}$ into German $\hat{\mathbf{e}}_{1/2}$ and use the En→Fr model to translate the other half of the English source $\mathbf{f}_{2/2}$ into French $\hat{\mathbf{e}}_{2/2}$, thus obtaining the pseudo-trilingual dataset $\{(\mathbf{f}_{1/2}, \hat{\mathbf{e}}^1_{1/2}, \mathbf{e}^2_{1/2}), (\mathbf{f}_{2/2}, \mathbf{e}^1_{2/2}, \hat{\mathbf{e}}^2_{2/2})\}$ that is as large as the original data. Pseudo datasets for De→En/Fr and En→Zh/Ja are generated similarly.

[19] https://dl.fbaipublicfiles.com/fasttext/supervised-models/lid.176.bin

Language pair	#Sentence(M)
En-De	11.52
En-Fr	33.90
De-Fr	5.58
En-Zh	9.93
En-Ja	7.22

Table 11: Statistics of WMT bilingual data used in pre-training experiments for multi-target translation.

For Bi-directional translation, we reuse the same trilingual data as described above. Pseudo data is generated by first training individual En→De$_{L2R}$ and En→De$_{R2L}$ Transformer models; we then follow the same procedure as above. The En→De$_{R2L}$ system is trained on En-De trilingual data with the German reference simply inverted. Pseudo data for the other language pairs is generated similarly.

B Details of Data for Code-switched Input Translation

We use the same WMT14 En-Fr data as in previous section to generate artificial code-switched sentences. These are obtained by randomly replacing small chunks in one sentence by their translation according to the following procedure. We first compute word alignments between parallel sentences using `fast_align`[20] (Dyer et al., 2013) in two directions, then apply a standard symmetrization procedure. Using the algorithm of Crego et al. (2005), we then identify bilingual phrase pairs $(\mathbf{f}, \mathbf{e})$ extracted from the symmetrized word alignments under the condition that all alignment links outgoing from words in $\mathbf{e}$ reach a word in $\mathbf{f}$, and vice-versa.

For each pair of parallel sentence, we first randomly select the primary language; then the number of substitutions r to perform using an exponential distribution as:

$$P(r = k) = \frac{1}{2^{k+1}} \quad \forall k = 1, \ldots, \text{rep}, \qquad (7)$$

where rep is the maximum number of replacements. We also make sure that the actual number of replacements never exceed half of either the original source or target sentence length, adjusting the actual number of replacements as:

$$n = \min(\frac{S}{2}, \frac{T}{2}, r), \qquad (8)$$

[20] https://github.com/clab/fast_align

Model	De-En	De-Fr	SIM	En-De	En-Fr	SIM	En-Zh	En-Ja	SIM	Avg BLEU	Avg SIM
base	33.0	26.2	90.09	28.8	38.1	91.77	28.4	13.7	81.99	28.0	87.95
multi	31.6	26.0	92.24	27.8	35.7	91.99	31.5	11.6	83.60	27.4 (-0.6)	89.28 (+1.33)
indep	34.3	26.8	90.78	29.8	38.7	92.00	29.1	14.3	83.22	28.8 (+0.8)	88.67 (+0.72)
dual	32.0	22.1	91.13	29.4	37.2	91.75	28.9	14.2	84.24	27.3 (-0.7)	89.04 (+1.09)
indep ps	33.5	26.7	91.05	29.1	38.7	92.52	28.8	13.8	83.37	28.4 (+0.4)	88.98 (+1.03)
dual ps	33.0	26.5	91.55	29.3	37.9	92.49	28.7	14.3	83.80	28.3 (+0.3)	89.28 (+1.33)
indep FT	37.0	29.5	91.98	32.1	42.0	92.53	32.0	16.5	84.74	31.5 (+3.5)	89.75 (+1.80)
dual FT	36.8	28.4	91.87	31.8	41.0	92.80	32.6	16.5	85.01	31.2 (+3.2)	89.89 (+1.94)
indep FT+ps	36.4	29.1	92.47	31.4	40.9	92.97	31.8	16.0	84.64	30.9 (+2.9)	90.03 (+2.08)
dual FT+ps	36.3	29.2	92.56	31.8	40.9	92.91	32.1	16.0	84.62	31.1 (+3.1)	90.03 (+2.08)

Table 12: BLEU and similarity scores of multi-target models on tst2015. Similarity scores (SIM) are computed as the cross-lingual similarity between the two target translations. Pseudo (ps) refers to models trained from scratch with synthetic reference data. FT indicates models fine-tuned from the pre-trained multilingual (multi) model. FT+ps refers to models fine-tuned using synthetic reference data.

where S and T are respectively the length of the source and target sentences. We finally choose uniformly at random r phrase pairs and replace these fragments in the primary language by their counterpart in the secondary language.

A shared vocabulary built with a joint BPE of 32K merge operations is used for CSW source as well as for English and French targets.

C Details of Data for Generating Translations of Varying Formalities

We reuse the data of Sennrich et al. (2016a).[21] The training data consists of OpenSubtitles2012 En-De data with 5.58M sentence pairs, out of which 0.48M of German reference are annotated as polite and 1.06M as impolite. The rest is deemed neutral. The annotation tool is based on the ParZu dependency parser[22] and an annotation script that is also released with the data. Polite/Impolite tags are based on an automatic analysis of the German side according to rules described in (Sennrich et al., 2016a). The test set that we use as development set is a random sample of 2000 sentences from OpenSubtitles2013. We use the testyou set as our main test set, which consists of 2000 random sentences also extracted from OpenSubtitles2013 where the English source contains a 2nd person pronoun *you(r(s(elf)))*.

We built shared vocabulary with a joint BPE of 32K merge operations. When fine-tuning the dual decoder models, we also randomly extract an equivalent number of neutral sentences as the polite and impolite ones, i.e. 1.54M. Reference of neutral sentences is thus identical for both polite and impolite targets. The overall fine-tuning data thus comprises 3.07M sentences.

D More Results of tst2015 for Multi-target Translation

Table 12 reports results for the multi-target translation experiments of Section 3 using the IWSLT tst2015, a setting that is also used in (He et al., 2021).

[21] http://data.statmt.org/rsennrich/politeness/

[22] https://github.com/rsennrich/ParZu