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 a single model.

Another common situation for MT applications is the multi-source / multi-target scenario, where source documents in language $S_i$ need to be published in several target languages $T^1_i, T^2_i, \ldots$. 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 $T^1_i$, which, once revised, can be used in conjunction with the original source to generate the translation into language $T^2_i$. The expected benefit of this approach is to facilitate word disambiguation.

An alternative, that we thoroughly explore here, is to simultaneously generate translations in $T^1_i$ and $T^2_i$, an approach termed multi-target translation by Neubig et al. (2015). While the same goal is achieved with a multilingual system translating independently in $T^1_i$ and $T^2_i$, 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 $T^1_i$ and $T^2_i$ 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 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 $H^1_l$ and $H^2_l$, the decoder cross-attention is computed as:

$$H^1_{l+1} = \text{Attention}(H^1_l, H^2_l),$$
$$H^2_{l+1} = \text{Attention}(H^2_l, H^1_l),$$

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 cross 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).

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

## 2 Architectures for Dual Decoding

### 2.1 Model and Notations

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

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

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

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

where $T = \text{max}(|e^1|, |e^2|)$, 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 $e^2_t$ is generated conditioned on $e^1_{<t+b}, e^2_{<t}$, thus reproducing the chained model of Le et al. (2020).

### 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 $Q$, key-value pairs $K$ and $V$. 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):

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right)V, \hspace{1cm} (4)$$

with $d_k$ the shared dimension of queries and keys. Note that $Q$, $K$ and $V$ 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. Alternate 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 $H^1_l$ and $H^2_l$, the decoder cross-attention is computed as:

$$H^1_{l+1} = \text{Attention}(H^1_l, H^2_l),$$
$$H^2_{l+1} = \text{Attention}(H^2_l, H^1_l),$$

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).
2.4 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.\footnote{In our experiments, the dual decoder was about twice as slow as the two independent ones.}

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.

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 decoders, implementing a controlled decoding where the output in $T^2_t$ not only translates the source, but does so in a way that is consistent with the reference translation in $T^1_t$. 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 $D = \{(f, e^1, e^2), i = 1 \ldots N\}$, the training maximizes the combined log-likelihood for the two target sequences:

$$L(\theta) = \sum_{D} \sum_{t=1}^{n_1} \log P(e_1^t | e_{<t}^1, e_{<t}^2, f, \theta) + \sum_{t=1}^{n_2} \log P(e_2^t | e_{<t}^2, e_{<t}^1, f, \theta),$$

where $\theta$ 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 $\{(f, e^1_i), i = 1 \ldots N^1\}$ and
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 (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.

<table>
<thead>
<tr>
<th></th>
<th>Original De</th>
<th>Original Fr</th>
<th>3-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>209522</td>
<td>236633</td>
<td>205397</td>
</tr>
<tr>
<td>Dev</td>
<td>2693</td>
<td>3083</td>
<td>2468</td>
</tr>
<tr>
<td>tst2014</td>
<td>1305</td>
<td>1306</td>
<td>1168</td>
</tr>
<tr>
<td></td>
<td>Original Zh</td>
<td>Original Ja</td>
<td>3-way</td>
</tr>
<tr>
<td>Train</td>
<td>235078</td>
<td>226384</td>
<td>213090</td>
</tr>
<tr>
<td>Dev</td>
<td>3064</td>
<td>3024</td>
<td>2837</td>
</tr>
<tr>
<td>tst2014</td>
<td>1297</td>
<td>1285</td>
<td>1214</td>
</tr>
</tbody>
</table>

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

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 (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.

For WMT data, we discard sentence pairs with invalid language tag as computed by fasttext language identification model (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

3.2 Experimental Settings

We implement the dual decoder model using fairseq14 (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.

---

2. Subject to some filtering to obtain a fully multi-parallel set of sentences.
4. For comparison with (He et al., 2021), we also report the results on tst2015 in Appendix D.
5. https://github.com/rsennrich/subword-nmt
7. Full experimental details are in Appendix A.
9. https://github.com/fxsjy/jieba
10. https://github.com/fxsjy/jieba
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 $f$ into targets $e_1$ and $e_2$, 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 $\{(f_1, e_1, e_2), (f_2, e_1, e_2)\}$ 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.

**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.

<table>
<thead>
<tr>
<th>Model</th>
<th>De-En</th>
<th>De-Fr</th>
<th>SIM</th>
<th>En-De</th>
<th>En-Fr</th>
<th>SIM</th>
<th>En-Zh</th>
<th>En-Ja</th>
<th>SIM</th>
<th>Avg BLEU</th>
<th>Avg SIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>32.6</td>
<td>24.8</td>
<td>89.28</td>
<td>28.1</td>
<td>38.8</td>
<td>91.34</td>
<td>22.2</td>
<td>13.6</td>
<td>81.97</td>
<td>25.5</td>
<td>87.53</td>
</tr>
<tr>
<td>multi</td>
<td>31.1</td>
<td>24.4</td>
<td>91.56</td>
<td>25.9</td>
<td>37.9</td>
<td>91.87</td>
<td>25.3</td>
<td>10.4</td>
<td>83.71</td>
<td>25.8 (+0.3)</td>
<td>89.05 (+1.52)</td>
</tr>
<tr>
<td>indep</td>
<td>33.8</td>
<td>25.4</td>
<td>90.04</td>
<td>29.1</td>
<td>39.8</td>
<td>91.63</td>
<td>22.6</td>
<td>14.8</td>
<td>83.16</td>
<td>26.3 (+0.8)</td>
<td>88.28 (+0.75)</td>
</tr>
<tr>
<td>dual</td>
<td>31.8</td>
<td>22.4</td>
<td>90.60</td>
<td>28.5</td>
<td>38.8</td>
<td>91.45</td>
<td>22.8</td>
<td>15.3</td>
<td>84.09</td>
<td>25.6 (+0.1)</td>
<td>88.71 (+1.18)</td>
</tr>
<tr>
<td>indep ps</td>
<td>33.4</td>
<td>26.1</td>
<td>90.51</td>
<td>28.5</td>
<td>39.6</td>
<td>91.98</td>
<td>22.7</td>
<td>14.3</td>
<td>83.58</td>
<td>26.2 (+0.7)</td>
<td>88.69 (+1.16)</td>
</tr>
<tr>
<td>dual ps</td>
<td>33.2</td>
<td>25.9</td>
<td>91.01</td>
<td>28.4</td>
<td>39.7</td>
<td>92.07</td>
<td>22.5</td>
<td>14.3</td>
<td>83.22</td>
<td>26.2 (+0.7)</td>
<td>89.00 (+1.47)</td>
</tr>
<tr>
<td>indep FT</td>
<td>37.1</td>
<td>28.6</td>
<td>91.52</td>
<td>30.1</td>
<td>42.3</td>
<td>92.25</td>
<td>26.5</td>
<td>17.1</td>
<td>84.86</td>
<td>30.3 (+4.8)</td>
<td>89.54 (+2.01)</td>
</tr>
<tr>
<td>dual FT</td>
<td>37.1</td>
<td>29.0</td>
<td>91.53</td>
<td>30.0</td>
<td>42.3</td>
<td>92.43</td>
<td>26.0</td>
<td>17.0</td>
<td>85.03</td>
<td>30.1 (+4.6)</td>
<td>89.66 (+2.13)</td>
</tr>
<tr>
<td>indep FT+ps</td>
<td>36.5</td>
<td>28.2</td>
<td>92.17</td>
<td>29.9</td>
<td>42.0</td>
<td>92.58</td>
<td>25.9</td>
<td>16.3</td>
<td>84.76</td>
<td>29.8 (+4.3)</td>
<td>89.87 (+2.34)</td>
</tr>
<tr>
<td>dual FT+ps</td>
<td>36.5</td>
<td>28.4</td>
<td>92.26</td>
<td>30.1</td>
<td>42.1</td>
<td>92.55</td>
<td>26.1</td>
<td>16.5</td>
<td>84.84</td>
<td>30.0 (+4.5)</td>
<td>89.88 (+2.35)</td>
</tr>
</tbody>
</table>

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 $\rightarrow$ De/Fr. Results using sequential decoding for one decoder are obtained in a second decoding pass using either automatic (auto) or reference (ref) translations.

<table>
<thead>
<tr>
<th>Model</th>
<th>En-De</th>
<th>En-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>dual</td>
<td>28.5</td>
<td>38.8</td>
</tr>
<tr>
<td>dual De+Fr auto</td>
<td>28.7</td>
<td>39.3</td>
</tr>
<tr>
<td>dual De+Fr ref</td>
<td>28.7</td>
<td>39.6</td>
</tr>
<tr>
<td>dual FT</td>
<td>30.0</td>
<td>42.3</td>
</tr>
<tr>
<td>dual FT De wait-3</td>
<td>30.2</td>
<td>42.4</td>
</tr>
<tr>
<td>dual FT Fr wait-3</td>
<td>30.4</td>
<td>42.6</td>
</tr>
<tr>
<td>dual FT De+Fr auto</td>
<td>30.0</td>
<td>42.5</td>
</tr>
<tr>
<td>dual FT De+Fr ref</td>
<td>30.0</td>
<td>42.4</td>
</tr>
</tbody>
</table>

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.

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 $\rightarrow$ 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-
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.

<table>
<thead>
<tr>
<th>Model</th>
<th>En-De Cons</th>
<th>En-Fr Cons</th>
<th>En-Zh Cons</th>
<th>En-Ja Cons</th>
<th>Avg BLEU</th>
<th>Avg Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>28.1</td>
<td>-</td>
<td>38.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>indep</td>
<td>29.1</td>
<td>54.7</td>
<td>39.4</td>
<td>65.8</td>
<td>22.2</td>
<td>51.3</td>
</tr>
<tr>
<td>dual</td>
<td>25.9</td>
<td>88.9</td>
<td>36.6</td>
<td>90.5</td>
<td>20.6</td>
<td>86.0</td>
</tr>
<tr>
<td>indep pseudo</td>
<td>29.0</td>
<td>65.7</td>
<td>39.9</td>
<td>73.3</td>
<td>22.9</td>
<td>62.0</td>
</tr>
<tr>
<td>dual pseudo</td>
<td>28.7</td>
<td>83.7</td>
<td>38.9</td>
<td>89.6</td>
<td>23.1</td>
<td>80.2</td>
</tr>
<tr>
<td>indep pseudo-dup</td>
<td>29.3</td>
<td>70.9</td>
<td>40.5</td>
<td>76.3</td>
<td>23.5</td>
<td>67.6</td>
</tr>
<tr>
<td>dual pseudo-dup</td>
<td>29.6</td>
<td>83.5</td>
<td>40.1</td>
<td>89.6</td>
<td>23.4</td>
<td>78.2</td>
</tr>
</tbody>
</table>

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.

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 pre-trained 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.

Table 6: Dual decoding for a CSW sentence.

<table>
<thead>
<tr>
<th>French</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>autrement dit, ils sortent du placard</td>
<td></td>
</tr>
<tr>
<td>In other words, they are getting out of the closet</td>
<td></td>
</tr>
</tbody>
</table>

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 pre-training 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>CSW-En</th>
<th>CSW-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>67.8</td>
<td>67.4</td>
</tr>
<tr>
<td>indep</td>
<td>67.7</td>
<td>67.3</td>
</tr>
<tr>
<td>dual</td>
<td>67.7</td>
<td>67.5</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Model</th>
<th>Exclusive</th>
<th>Both</th>
<th>Punctuations</th>
<th>Lost</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>81.56</td>
<td>8.10</td>
<td>10.34</td>
<td>0.72</td>
</tr>
<tr>
<td>base</td>
<td>79.14</td>
<td>8.85</td>
<td>11.29</td>
<td>0.72</td>
</tr>
<tr>
<td>indep</td>
<td>78.86</td>
<td>9.13</td>
<td>11.35</td>
<td>0.67</td>
</tr>
<tr>
<td>dual</td>
<td>78.90</td>
<td>9.17</td>
<td>11.32</td>
<td>0.61</td>
</tr>
</tbody>
</table>

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). 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. Using this data, we generate tri-parallel data as follows. We first train a tag-based model.
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 test set, which contains 2k sentences with a second-person pronoun you(rself)) 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Tag</th>
<th>neutral</th>
<th>pol</th>
<th>imp</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref</td>
<td></td>
<td>438</td>
<td>525</td>
<td>1037</td>
<td>-</td>
</tr>
<tr>
<td>pre-train</td>
<td>none</td>
<td>1914</td>
<td>16</td>
<td>70</td>
<td>17.7</td>
</tr>
<tr>
<td></td>
<td>pol</td>
<td>479</td>
<td>1518</td>
<td>3</td>
<td>20.9</td>
</tr>
<tr>
<td></td>
<td>imp</td>
<td>22</td>
<td>0</td>
<td>1978</td>
<td>24.1</td>
</tr>
<tr>
<td></td>
<td>oracle</td>
<td>551</td>
<td>406</td>
<td>1043</td>
<td>30.2</td>
</tr>
<tr>
<td>Dec</td>
<td>neutral</td>
<td>528</td>
<td>1470</td>
<td>2</td>
<td>21.0</td>
</tr>
<tr>
<td></td>
<td>pol</td>
<td>582</td>
<td>82</td>
<td>0</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>imp</td>
<td>514</td>
<td>1457</td>
<td>2</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>0</td>
<td>1903</td>
<td>24.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>seq imp</td>
<td>531</td>
<td>1467</td>
<td>2</td>
<td>21.2</td>
</tr>
<tr>
<td></td>
<td>seq pol</td>
<td>84</td>
<td>0</td>
<td>1916</td>
<td>24.4</td>
</tr>
</tbody>
</table>

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


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


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.

<table>
<thead>
<tr>
<th>Original De</th>
<th>Original Fr</th>
<th>3-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>209522</td>
<td>236653</td>
</tr>
<tr>
<td>Dev</td>
<td>2693</td>
<td>3083</td>
</tr>
<tr>
<td>tst2014</td>
<td>1305</td>
<td>1306</td>
</tr>
<tr>
<td>tst2015</td>
<td>1080</td>
<td>1210</td>
</tr>
</tbody>
</table>

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.

For Bi-directional translation, we reuse the same trilingual data as described above. Pseudo data is generated by first training individual En→De\textsubscript{L2R} and En→De\textsubscript{R2L} Transformer models; we then follow the same procedure as above. The En→De\textsubscript{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\textsubscript{align}\textsuperscript{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 \((f, e)\) extracted from the symmetrized word alignments under the condition that all alignment links outgoing from words in \(e\) reach a word in \(f\), 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:

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

where \(\text{rep}\) 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:

\[
n = \min\left(\frac{S}{2}, \frac{T}{2}, r\right),
\]

---

\(^{19}\)https://dl.fbaipublicfiles.com/fasttext.supervised-models/lid.176.bin

\(^{20}\)https://github.com/clab/fast\textsubscript{align}
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). 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 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).