Discriminative Reranking for Neural Machine Translation

Ann Lee **Michael Auli** **Marc'Aurelio Ranzato**
Facebook AI Research
{annl,michaelauli,ranzato}@fb.com

Abstract

Reranking models enable the integration of rich features to select a better output hypothesis within an n-best list or lattice. These models have a long history in NLP, and we revisit discriminative reranking for modern neural machine translation models by training a large transformer architecture. This takes as input both the source sentence as well as a list of hypotheses to output a ranked list. The reranker is trained to predict the observed distribution of a desired metric, e.g. BLEU, over the n-best list. Since such a discriminator contains hundreds of millions of parameters, we improve its generalization using pre-training and data augmentation techniques. Experiments on four WMT directions show that our discriminative reranking approach is effective and complementary to existing generative reranking approaches, yielding improvements of up to 4 BLEU over the beam search output.

1 Introduction

Reranking models take a number of different output hypotheses generated by a baseline model and select one hypothesis based on more powerful features. Before the recent re-emergence of neural networks, these models have been well studied for several NLP tasks including parsing (Charniak and Johnson, 2005; Collins and Koo, 2005) and statistical machine translation (Och et al., 2004; Shen et al., 2004).

Traditional statistical models (SMT) based on n-gram counts made very strong independence assumptions where features would only capture very local context information to avoid sparsity and poor generalization. A large n-best list produced by these models would then be passed to a discriminatively trained reranker which leverages features engineered to capture more global context (Och et al., 2004) yielding significant improvements to the quality of the translations.

On the other hand, modern neural models (NMT) make much weaker independence assumptions because predictions of standard sequence-to-sequence models depend on the entire source sentence as well as the target prefix generated. However, reranking may still be beneficial for two reasons: First, NMT systems are subject to exposure bias (Ranzato et al., 2016), i.e., models are never exposed to their own generations at training time, while a reranking model has been trained on model outputs. Second, beam search with autoregressive models uses the chain rule to sum individual token-level probabilities to obtain a target sequence probability. However, individual probabilities are based on a limited amount of target context, while a reranking model can condition on the entire target context. Indeed, recent *generative* reranking approaches applied to NMT, such as Noisy-Channel Decoding (NCD, Yee et al. 2019) which leverages a pre-trained language model and a backward model, show strong improvements over beam search outputs, as demonstrated in recent WMT evaluations (Ng et al., 2019).

In this paper, we explore whether training large transformer models using the reranking objective can further improve performance. Our model, dubbed `DrNMT`, takes as input the entire source sentence and an n-best list of output hypotheses to predict a distribution of sentence-level evaluation scores, such as BLEU.[1] This setup is similar to earlier work with SMT, except that the baseline model is an NMT model and the reranker is a big transformer architecture as opposed to a log-linear model on top of discrete or human engineered features.

Unfortunately, optimizing for the task of interest does not always lead to better performance. Overfitting to the training set is a potential concern, as the

[1] Our approach is general and enables optimizing any user-specified metric, or combinations thereof.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 7250–7264
August 1–6, 2021.

reranker has hundreds of millions of parameters yet it receives only one gradient and weight update per source/target sentence pair as opposed to one per token as for standard NMT models. In our work, we mitigate overfitting in two ways. First, we leverage the success of pre-training by finetuning masked language models (MLM; Devlin et al. 2019) which initializes the model with features trained on much more training data. Second, we augment the original dataset with back-translated data (BT; Sennrich et al. 2016).

Experiments show that `DrNMT` can match the performance of a strong NCD baseline and that their combination leads to further improvements as measured by BLEU, TER and also human evaluation.

2 Related Work

Our method is inspired by the seminal work of Shen et al. (2004) and Och et al. (2004) who introduced and popularized discriminative reranking to SMT. Besides using a weaker MT system to generate the n-best list, these works relied on a linear discriminator trained on human-designed features as opposed to a transformer taking the raw source sentence and hypothesis.

Most work using NMT has focused on generative reranking methods (Liu et al., 2018; Imamura and Sumita, 2017; Wang et al., 2017), where the reranker's parameters are optimized using a criterion which is different from the metric of interest. For instance, Yu et al. (2017); Yee et al. (2019) perform noisy-channel decoding where hypotheses are scored by linearly combining the output of the forward model, a target-side language model and a backward model which scores the source sentence given the hypothesis. These methods have shown remarkable improvements over the output of beam decoding, despite not being trained for the reranking task (except for the two or three hyperparameters of the linear combination of scores which are tuned on a validation set). Another approach belonging to this class of methods is the one proposed by Salazar et al. (2019), which employs the scores from a masked language model (MLM). While this method employs a transformer architecture, it is still not trained for the task of interest.

To the best of our knowledge, there is only concurrent work by Naskar et al. (2020) which attempts at training discriminatively a reranker for NMT. They use a pair-wise margin loss on hypotheses sampled from the NMT, while we learn to rank the full n-best list produced by beam. Their experiments also show that the reranker performs better when directly conditioned on the source sentence. However, they do not compare nor combine their method with NCD like we do. Both their work and our work are however an extension of Deng et al. (2020), who proposed to train a discriminator to improve neural language modeling.

There is also a large body of literature on different ways to combine SMT and NMT by using one to rerank the other, since SMT is generally better at adequacy while NMT is better at fluency. For instance, Auli and Gao (2014) uses an RNN discriminator to rerank the n-best list produced by a phrase-based SMT. Instead, Ehara (2017) does the opposite, using an SMT discriminator to rerank an n-best list produced by an NMT.

Finally, our work is also related to recent attempts at using adversarial training to improve MT (Wu et al., 2018; Zhang et al., 2018). Unlike these approaches our method is much simpler because we do not update the parameters of the MT system generating the hypotheses. Moreover, our discriminator is trained to predict the distribution of desired metric and it is used at decoding time to rerank, while GAN-based MT would only retain the generator.

3 Model

Given a source sentence x, an NMT model generates a set of hypotheses $\mathcal{U}(x) = \{u_1, u_2, ..., u_n\}$ in the target language. The goal of this work is to learn a reranker that produces higher scores for hypotheses of better quality, as defined in terms of a user-specified metric $\mu(u, r)$ such as BLEU (Papineni et al., 2002a), where quality is measured with respect to a reference r.

As illustrated in Figure 1, our reranker is a transformer architecture which takes as input the concatenation of the source sentence x and hypothesis $u \in \mathcal{U}(x)$. The architecture includes also position embeddings and language embeddings, to help the model represent tokens that are shared between the two languages (Conneau and Lample, 2019). The final hidden state corresponding to the start of sentence token ($\langle s \rangle$) serves as the joint representation for (x, u); let us denote this feature vector as $z \in \mathbb{R}^d$. The reranker associates a scalar score $o \in \mathbb{R}$ to (x, u) by applying a one hidden layer

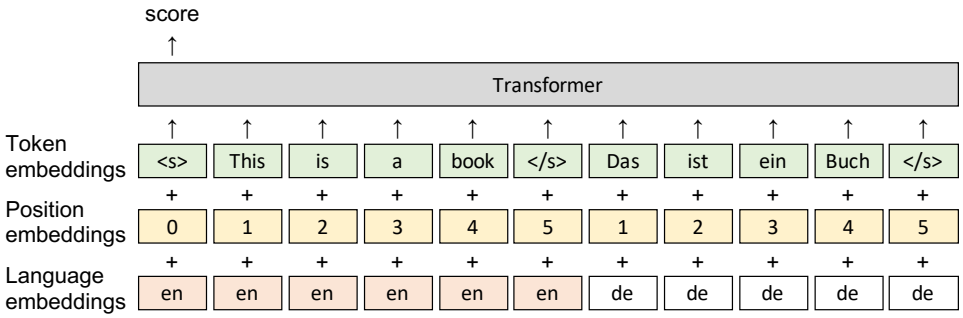

Figure 1: Illustration of DrNMT, a pre-trained transformer architecture which takes as input both the source sentence as well as a hypothesis and outputs a scalar score. DrNMT is trained to output scores which reflect the distribution of sentence-level scores according to a user-specified metric over an n-best list.

neural network with d tanh hidden units to z, as default in the design of the "classification head" of RoBERTa (Liu et al., 2019). The parameters of the reranker are denoted by θ and include the parameters of the transformer, all the embeddings and also the top projection block mapping the feature vector to the scalar score. Each hypothesis u_i in the set $\mathcal{U}(x)$ is therefore processed independently and yields a score o_i.

4 Training and Inference

We train the reranker discriminatively, hence the name DrNMT for Discriminative Reranker for NMT, by minimizing the KL-divergence between the target distribution and the model output distribution, $D_{KL}(p_T||p_M)$ (Cao et al., 2007). For each x, the model output distribution is a softmax over all n hypotheses in the n-best list:

$$p_M(u_i|x;\theta) = \frac{\exp(o_i(u_i|x;\theta))}{\sum_{j=1}^{n}\exp(o_j(u_j|x;\theta))}, \quad (1)$$

where we made explicit that the score o_j is conditioned on the input x and parameter vector θ. Notice that we do not enforce any additional factorization. In particular, we do not assume that the score is computed auto-regressively.

The target distribution is defined as a normalized distribution of the end metric $\mu(u_i, r)$ which we assume to improve as it takes on larger values:

$$p_T(u_i) = \frac{\exp(\mu(u_i, r)/T)}{\sum_{j=1}^{n}\exp(\mu(u_j, r)/T)}, \quad (2)$$

where T is the temperature to control the smoothness of the distribution. In practice, we apply a min-max normalization on μ. We subtract each value by the minimum in the hypothesis set, and divide the result by the difference between the maximum and the minimum value, so that the best hypothesis scores 1 and the worst 0. This helps the optimization as it reduces the variance of the gradients, as pointed out by Edunov et al. (2018).

The parameters of DrNMT are then learned by minimizing the KL divergence over the training dataset. For a given training example, we have:

$$\mathcal{L}(\theta) = -\sum_{j=1}^{n} p_T(u_j)\log p_M(u_j|x;\theta). \quad (3)$$

We minimize this loss over the training set by stochastic gradient descent using standard back-propagation of the error, since all terms are differentiable. In order to alleviate overfitting, we employ dropout regularization (Srivastava et al., 2014), we pre-train the model (Conneau et al., 2019) and we also perform data augmentation by training on back-translated data (BT) (Sennrich et al., 2016). See §5.3 for details.

At test time, generation proceeds by first having the NMT generate the n-best list, and then by applying the reranker to select the best hypothesis. Since the score of the forward model is also available, unless otherwise specified we rerank using a weighted combination of both; this is dubbed as DrNMT. In the experiments we also report results by adding all the other scores from NCD, namely the backward model score and the language model score. We denote this variant by "DrNMT + NCD". Whenever we combine scores from various models we tune the additional hyper-parameters controlling the weighted combination by random search on the validation set (Yee et al., 2019).

5 Experimental Setup

In this section we describe the datasets, baselines and model details.

5.1 Datasets

We experiment on four language pairs: German-English (De-En), English-German (En-De), English-Tamil (En-Ta) and Russian-English (Ru-En). For training on De-En and En-De, we use NewsCommentary from WMT'19 (Barrault et al., 2019) and NewsCrawl2018 for the parallel dataset and target side monolingual data, respectively. We validate on newstest2014 and newstest2015, and test on newstest2016, 2017, 2018 and 2019. For En-Ta, we use all bitext and monolingual data shared by the WMT'20 news translation task for training, and the officially released development and test sets for validation and testing purposes. For Ru-En, we use all the parallel data from WMT'19 (Barrault et al., 2019) and NewsCrawl2018 as the monolingual dataset for training, validate on newstest2015 and 2016, and test on newstest 2017, 2018 and 2019.

We follow the steps in Ng et al. (2019) for data preprocessing, including sentence deduplication, language identification filtering on all bitext and monolingual data (Joulin et al., 2017) and in-domain filtering (Moore and Lewis, 2010) on Tamil CommonCrawl data. Table 1 shows the resulting size of each dataset. For the base NMT models, we learn 30K byte-pair encoding (BPE) units for De-En and En-De, 20K BPE units for En-Ta and 24K BPE units for Ru-En separately, using the `sentencepiece` toolkit (Kudo and Richardson, 2018). All systems are evaluated using SACREBLEU (Post, 2018).

5.2 Baselines

We use the Transformer (Vaswani et al., 2017) architecture and train MT models using bitext data only. These are the models that generate the n-best list, and which serve also as a lower bound for the performance of `DrNMT`. BT data is generated from beam decoding with beam size equal to 5. Since the bitext data of En-Ta originates from seven different sources, we prepend dataset tags to each source sentence to indicate the origin (Kobus et al., 2017). We do not prepend any tags on the validation and test sets when decoding, as this choice worked best during cross-validation. In general and for each language pair, we tune the model architecture and all hyper-parameters on the validation set.

	De-En	En-De	En-Ta	Ru-En
bitext				
training	326K	326K	621K	28.9M
validation	5.2K	5.2K	2K	5.8K
test	11K	11K	1K	8K
monolingual	17M	37M	27M	17M

Table 1: Number of sentences in each dataset used in the experiments after pre-processing.

In addition to beam decoding, we consider two reranking baselines. First, we consider the method recently introduced by Salazar et al. (2019). In its simplest formulation, this takes a pre-trained masked language model (MLM) on the target side, and iteratively masks one word of the hypothesis at the time and aggregates the corresponding scores to yield a score for the whole hypothesis. Then, this score is combined with the score of the forward model to rerank the n-best list; this is dubbed as "fw + MLM". We also have a version of MLM which is tuned on our target side monolingual dataset; we dub this "fw + MLM-ft".

Finally, we consider reranking using noisy channel decoding (NCD; Yee et al. 2019). NCD reranks by taking a weighted combination of three scores: the forward model score, the score of a target-side language model (LM), and the score of a backward model. A length penalty is then applied on the combined score. The weights and the length penalty are tuned on the validation set via random search. All LMs are transformers with 16 blocks, 16 attention heads and embedding size 1024. They are trained on the target side monolingual data only.

5.3 Setting Up `DrNMT`

We use XLM-R_{Base}[2] (Conneau et al., 2019), a transformer-based multilingual MLM trained on more than 2.5T of of filtered CommonCrawl data in 100 languages, including En, De, Ta and Ru, as the pre-trained model for `DrNMT`. The same model is also used in the MLM baseline described in §5.2. The XLM-R_{Base} model consists of 12 transformer blocks, 12 attention heads, embedding size 768 (270M params) and has a vocabulary size of 250K BPE units. As each training sample of XLM-R only contained one single language, we further enhance the model with two language embeddings,

[2] https://github.com/pytorch/fairseq/tree/master/examples/xlmr

	De-En		En-De		En-Ta		Ru-En	
BLEU	valid	test	valid	test	valid	test	valid	test
beam (fw)	24.7	27.7	23.1	26.6	8.8	6.0	33.5	34.3
+ MLM (Salazar et al., 2019)	25.7	28.7	23.5	27.1	8.8	5.8	33.8	34.8
+ MLM-ft (Salazar et al., 2019)	25.8	28.8	23.7	27.5	8.8	5.8	33.9	35.0
+ LM	26.3	29.2	24.3	28.5	9.4	6.2	34.6	35.8
NCD (Yee et al., 2019)	27.2	30.9	24.8	29.1	9.7	6.3	35.3	36.8
DrNMT	27.6	31.5	24.7	29.0	9.7	6.4	35.3	37.1
+ NCD	**27.9**	**31.8**	**25.1**	**29.7**	**10.0**	**6.5**	**35.7**	**37.3**
oracle BLEU	33.3	37.4	31.4	35.9	13.6	9.5	45.3	47.0

Table 2: Validation and test BLEU with beam size 50. Results for De-En and En-De are averaged from newstest2014 and 2015 for validation and newstest2016, 2017, 2018 and 2019 for test. The results for Ru-En are averaged from newstest2015 and 2016 for validation and newstest2017, 2018 and 2019 for test.

initialized from random, to indicate the source and target languages for the reranker.

We perform beam decoding on both bitext and BT data using the baseline MT models to generate n-best lists with 50 hypotheses. We combine n-best lists from both bitext and BT as training data for the rerankers for De-En, En-De and En-Ta, and use only BT data for Ru-En. We train DrNMT with batch size 512, use Adam (Kingma and Ba, 2015) and early-stop when the validation performance does not improve after 12K parameter updates. All hyper-parameters, including learning rate, number of warmup steps, dropout rate, etc., are tuned on the validation set. All models are implemented and trained using fairseq (Ott et al., 2019)[3].

6 Results

In this section we report the main findings of our work. When optimizing for BLEU as metric, the performance of DrNMT and baselines for De-En, En-De, En-Ta and Ru-En is summarized in Table 2. The findings are similar across the four language directions. We therefore focus the discussion on the De-En test set results.

First, we notice that all methods improve over the beam search output with gains ranging from 1.0 to 4.1 BLEU. However, there may be still room for improvement as the oracle performance suggests. The oracle is computed by selecting the best hypotheses based on BLEU with respect to the human reference. Of course, the oracle may be not achievable because of uncertainty in the translation task.

[3]Code for reproducing the results can be found at: https://github.com/pytorch/fairseq/tree/master/examples/discriminative_reranking_nmt

Second, Salazar et al. (2019)'s method, particularly the version fine-tuned on the in-domain training dataset, improves upon beam by 1.1 BLEU points. However, the improvement over beam is not as large as with NCD, which improves upon beam by 3.2 BLEU points, suggesting that among the non-discriminative reranking methods NCD performs the best.

Third, DrNMT performs on par (En-Ta, En-De and Ru-En) or better (De-En) than NCD, showing that discriminative reranking can be very competitive. Note, that the reranker requires only one additional forward pass through the hypotheses generated by beam, while NCD requires two forward passes (one for the LM and one for the backward MT model). Therefore, our reranker works at least as well as NCD while requiring roughly half of the compute.

Fourth, the discriminative reranker and NCD are complementary to each other, since combining both achieves the best performance overall across the three language directions, with gains between 0.9 BLEU (De-En) and 0.2 (En-Ta) compared to NCD, and an overall gain between 4.1 BLEU (De-En) and 0.5 (En-Ta) compared to the beam baseline.

Fifth, the gain brought by discriminative reranking can be better appreciated by comparing "fw + LM" and DrNMT, as the major difference between the two approaches is the objective function used for training them (generative language modeling instead of prediction of the distribution of BLEU scores). We can see that in all cases, discriminative reranking yields better translations, with gains between 0.2 and 2.3 BLEU points depending on the language direction.

Finally, we notice that En-Ta is a difficult lan-

	valid		test	
	BLEU	TER	BLEU	TER
beam	24.7	60.9	27.7	58.0
DrNMT (B)	27.6	57.7	31.5	54.1
+ NCD	**27.9**	57.9	**31.8**	54.2
DrNMT (T)	27.0	57.3	30.7	53.5
+ NCD	27.3	**57.0**	31.1	**53.4**

Table 3: Average validation and test BLEU and TER on WMT'19 De-En with beam size 50 from rerankers trained with different metrics (B: BLEU, T: TER).

guage pair, in which the baseline NMT is weak and none of the reranking approaches work nearly as well as in the other language directions. The difference between validation and test BLEU scores suggests also a certain degree of overfitting to the validation set. Despite this, our reranker still yields the largest improvement over beam. Appendix B shows similar trends when test performance is measured in terms of translation error rate (TER) (Snover et al., 2006), showing that DrNMT is not particularly overfitting to the training metric.

Human evaluation: We randomly sample 750 sentences from the De-En test sets and collect human ratings. We perform A/B testing, where a rater can see the source sentence together with translated sentences from two systems. We conduct two rounds of human evaluation by comparing the proposed "DrNMT + NCD" vs. "beam", and "DrNMT + NCD" vs. "NCD". For each sentence, we collect three ratings (between 0 to 100) and average the scores, treating sentences with a score difference less than 5 as equally good. Out of the 750 sentences, our proposed method generates better translation than beam on 149 sentences and is worse on 82 sentences, and it performs better than NCD on 123 sentences and worse on 108 sentences, corroborating the gains observed when measuring with BLEU.

Next we show that DrNMT works with other user-specified metrics, study how performance varies with the number of hypotheses and perform several ablation studies to better understand its critical components.

6.1 Optimizing for a Different Metric

In order to validate the generality of DrNMT, we consider as metric μ the opposite of TER, so that larger values indicate better translation quality.

Table 3 shows validation and test performance in terms of both BLEU and TER when optimizing for either one of the two metrics. While the two metrics are correlated, the best results are achieved when optimizing for the metric used at test time.

6.2 Varying the Number of Hypotheses

We examine the effect of training the reranker with different sizes of the n-best list, $\mathcal{U}(x)$. Even though we fix the n-best list size at training time, we can apply the reranker on n-best lists of different sizes at test time. Figure 2 shows the performance of DrNMT on De-En validation sets from four rerankers trained with 5, 10, 20 and 50 hypotheses, respectively.

As the size of the n-best list during test time increases, the performance of all rerankers and NCD improve. On the other hand, the performance of beam decoding starts to saturate early at beam size 10. A reranker trained with 50 hypotheses gives a 1.4 BLEU improvement over beam decoding when beam size is only 5 at test time, and the improvement increases to 3.4 BLEU as we increase the beam size to 200 at test time. DrNMT consistently perform better than or equally well as NCD in all training and testing scenarios.

Interestingly, a reranker trained with more hypotheses performs better than one trained with fewer hypotheses, regardless of the beam size used at test time. For instance, when the beam size is 20 at test time, the reranker trained with beam 50 improves over beam by 2.3 BLEU points, while the one which was trained with 20 like at test time, improves by 2.2 BLEU points.

To our surprise, a reranker trained with only 5 hypotheses can still yield a 3.2 BLEU gain compared with beam decoding when used to rerank 200 hypotheses during test time, indicating that the reranker suffers little from the mismatch between training and testing conditions. As a result, depending on available compute resources, one can decide to set the number of hypotheses to the largest value possible to get better test time performance with larger n-best lists, while being robust to the particular choice used at training time.

6.3 Ablation Study

We report an ablation study by probing all major design choices made. We train DrNMT by optimizing BLEU and evaluate it on the validation set of the De-En task using 50 hypotheses both at training and test time. Table 4 summarizes all the results.

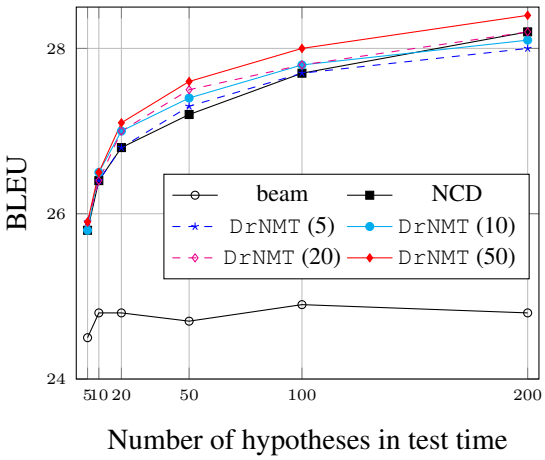

Figure 2: BLEU on the validation set of De-En of rerankers trained with n equal to 5, 10, 20 or 50 hypotheses (denoted by "DrNMT (n)") and NCD when reranking using different numbers of hypotheses at test time (x-axis).

	valid
proposed	27.6
- pre-training	26.8
- source sentence	27.4
- normalization	27.2
- BT data	25.6
6 layers	27.1
3 layers	26.7

Table 4: Ablation study on the various design choices of the proposed approach. All results are evaluated on the De-En validation set.

Pre-training: We investigate the importance of pre-training by comparing with a reranker of the same size initialized with random weights. Table 4 shows that a randomly initialized reranker performs significantly less well, with a decrease of 0.8 BLEU. In addition to lower performance, a randomly initialized reranker also trains more slowly, by requiring $1.6\times$ more weight updates compared to the pre-trained reranker to converge. This corroborates our choice to pre-train, as the reranking task is fairly related to the pre-training task and we lack sufficient labeled data to train such a large model from scratch. Notice that our pre-trained reranker trains for at most two passes over the data before starting to overfit to its training set.

Source sentence: When comparing "fw + LM" against DrNMT to assess the impact of training discriminatively, we did not take into account a confounding factor which is the fact that the LM does not attend over the source sentence. Indeed, Salazar et al. (2019) score hypotheses without taking into account the source sentence. What is the gain brought by considering also the source sentence? To answer this question we compare our reranker with a reranker that takes as input only the hypotheses. As shown in Table 4, including the source sentences achieves a small gain of 0.2 BLEU.

Normalization: We apply minmax normalization and set $T = 0.5$ when computing the target distribution in the training objective, so that for every source sentence, the range of the BLEU scores of its hypotheses is between 0 and 2. This choice yields a 0.4 BLEU improvement compared to a reranker trained with the raw BLEU scores.

Training data: So far we've been training the reranker with both bitext and BT data. In Table 4, we see that training the reranker with only bitext data deteriorates the model's performance by 2 BLEU points. The model starts overfitting after 15 passes over the small bitext (around 9,000 parameter updates). Incorporating the BT data helps alleviate this issue. The model achieves the best validation performance after 1.9 passes over the combination of bitext and BT data (around 63,000 parameter updates).

Model size: We explore building the reranker using only the first few layers of the XLM-R_{Base} model. Since beam hypotheses often differ only locally on isolated phrases, one may wonder whether more local features, as those produced by a shallower reranker may work better. Moreover, reducing the model capacity may help preventing overfitting. Compared with either only three or six transformer blocks, Table 4 shows that deeper and bigger models work better, despite being more prone to overfitting and despite capturing more global information about their input.

6.4 Other Training and Model Variations

We conclude our empirical evaluation by investigating how reranking works on top of baseline NMT models trained with back-translation, and by reporting two variations of model architectures. As before, we report results on the validation set of the De-En task with n-best list of size 50, using BLEU as metric.

MT trained with bitext+BT: Would the gains brought by the reranker carry over when this is ap-

	valid
beam (fw)	31.6
+ MLM (Salazar et al., 2019)	32.6
+ MLM-ft (Salazar et al., 2019)	32.6
+ LM	33.1
NCD (Yee et al., 2019)	33.3
DrNMT	33.1
+ NCD	33.6

Table 5: Reranking the output of a baseline trained with back-translation.

	valid
encoder + causal decoder	26.8
bi-directional (proposed)	27.6

Table 6: Effect of a causal vs. non-causal reranker.

	valid
set-level	27.6
hypothesis-level (proposed)	27.6

Table 7: Reranking with features computed over the entire n-best list (set-level reranking) vs. features from just the current hypothesis.

plied on the n-best list produced by a baseline NMT model trained with back-translation? As shown in Table 2 the beam baseline on validation was at 24.7 BLEU, while if we train the NMT by adding back-translated data, BLEU increases to 31.6 (Table 5). In this case, we train the reranker using hypotheses generated by the more powerful NMT model trained with back-translated data. From Table 5, we can see that DrNMT gives 1.5 BLEU improvement over the beam decoding baseline, and combining NCD and reranker gives an additional gain of 0.5 BLEU, which is less than what we reported in Table 2 but still confirming the overall finding of discriminative reranker and NCD performing similarly while being complementary to each other.

Causal vs. bidirectional: As the complete hypothesis is available during reranking, the architecture of our reranker is bidirectional as it conditions on the whole sentence. This contrasts with how the baseline NMT model generates hypotheses and how it scores them with beam which leverages an auto-regressive decomposition. Here we explore the importance of joint modeling and consider an alternative reranker which consists of an encoder and a *causal* decoder, and which is therefore initialized from the base NMT generating the n-best list. Given a source sentence and a hypothesis as input, the output of the decoder is a $T \times d$ matrix (notice that hidden states are causal), where T is the number of tokens of the hypothesis, and d is the hidden dimension. We average the output across position to obtain a d-dimensional representation and apply the same one-hidden layer neural network to obtain a reranking score. Table 6 shows that our bidirectional architecture outperforms the causal architecture by 0.8 BLEU.

Set reranker: While our training objective considers the full set of hypotheses of each source sentence, the reranker scores each pair of (x, u_i) in isolation; it never compares hypotheses directly. We therefore explore an architecture that computes cross-hypothesis features. In the original reranker architecture, the model produces a d-dimensional representation for each (x, u_i). We add another transformer block that computes self-attention across the set of n representations for $\{(x, u)|u \in \mathcal{U}(x)\}$. We then apply the one hidden layer projection block to map each d dimensional vector to a single score as before, yielding n scores for reranking. This design enables the model to have set-level information during reranking, and thus the scoring has to be performed on the full set at once. Table 7 shows that these two model variants perform the same, suggesting that set level representations may need to be captured at a lower layer of the transformer. We leave this avenue of exploration for future work.

7 Conclusions

Reranking is effective for both SMT and NMT. Inspired by work done almost two decades ago (Shen et al., 2004; Och, 2003), we studied discriminative reranking for NMT and found that it performs at least as well as the strongest generative reranking method we are aware of, namely noisy channel decoding (NCD) (Yee et al., 2019) - as long as care is taken to alleviate overfitting.

There is a subtle trade-off between improvements stemming from optimizing the end metric and addressing exposure bias on the one hand, and poor generalization and sample inefficiency of discriminative training on the other hand. In this study we regularize the reranker by using dropout, by pre-training on large corpora and by performing data augmentation.

Empirically, we found that NCD and our discrim-

inative reranker are complementary to each other, yielding sizeable improvements over each other and the beam baseline. Our reranker is computationally less demanding than NCD, since it consists of a single model while NCD requires scoring using two additional models. Our reranker is also robust to the choice of the size of the n-best list and other hyper-parameters settings.

In the future we plan to investigate better ways to alleviate sample inefficiency, as well as to design more effective architectures to score at the set level.

Acknowledgments

We would like to thank Peng-Jen Chen for his guidance on training NMT systems, and Sergey Edunov for his advice on setting up the human evaluation.

References

Michael Auli and Jianfeng Gao. 2014. Decoder integration and expected BLEU training for recurrent neural network language models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 136–142.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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

Terumasa Ehara. 2017. SMT reranked NMT. In *Workshop on Asian Translation*.

Kenji Imamura and Eiichiro Sumita. 2017. Ensemble and reranking: Using multiple models in the NICT-2 neural machine translation system at WAT2017. In *Workshop on Asian Translation*.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Catherine Kobus, Josep M Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. A comparable study on model averaging, ensembling and reranking in NMT. In *International Conference on Natural Language Processing and Chinese Computing*, pages 299–308. Springer.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the Association for Computational Linguistics 2010 Conference Short Papers*, pages 220–224.

Subhajit Naskar, Amirmohammad Rooshenas, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2020. Energy-based reranking: Improving neural machine translation using energy-based models. *arXiv:2009.13267*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David A Smith, Katherine Eng, et al. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2019. Masked language model scoring. *arXiv preprint arXiv:1910.14659*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Association for Computational Linguistics*.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. *arXiv:1704.06933*.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *International Conference on Learning Representations, ICLR 2017*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Bidirectional generative adversarial networks for neural machine translation. In *Proceedings of the 22nd conference on computational natural language learning*, pages 190–199.

A Training details

A.1 MT model

We build the baseline MT models in Table 2 following the Transformer big architecture (Vaswani et al., 2017) with 6 layers, embedding size 1024 and 16 attention heads. Table 8 shows the additional hyper-parameters that we tune on the validation set for the best performing models of each language direction. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 0.00000001$, and apply an inverse square root learning rate schedule with 4000 warmup steps. We train for 200 epochs for De-En, En-De and En-Ta, and 100K updates for Ru-En, and select the best checkpoint based on validation loss.

A.2 LM

For all LMs, we use 16 transformer layers, embedding size 1024, feed-forward network embedding size 4096 and 16 attention heads. We optimize with NAG with learning rate 0.0001 and a cosine learning rate schedule with 16K warmup steps. All models are trained on 32 GPUs for a maximum of 984K steps, and the best checkpoint is selected based on validation loss.

A.3 DrNMT

We train DrNMT using Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 0.000001$, and apply a polynomial learning rate decay schedule with 8000 warmup steps for De-En, En-Ta, and Ru-En, and 16K warmup steps for En-De. We use a learning rate of 0.00005 and dropout 0.2 for De-En, En-Ta, and Ru-En, and a learning rate of 0.00001 and dropout 0.1 for En-De.

B TER results

Table 9 summarizes the average validation and test TER (Snover et al., 2006) of DrNMT trained with BLEU (Papineni et al., 2002b) scores. Table 10, Table 11, and Table 12 show TER of each validation and test set for De-En, En-De and Ru-En, respectively. Note that for En-Ta we only have one validation and one test set.

C BLEU results

Table 13, Table 14, and Table 15 show the performance of DrNMT, trained and evaluated on BLEU, on each validation and test set for De-En, En-De and Ru-En, respectively. The average validation and test BLEU scores of each language pair are reported in the main paper in Table 2.

D Examples

Table 16 and Table 17 show examples of translation from NCD and DrNMT + NCD.

	# params	ffn embed size	learning rate	dropout	label smoothing	max tokens per GPU	# GPUs
De-En	207M	4096	0.0007	0.3	0.2	4000	4
En-De	207M	4096	0.0003	0.4	0.3	4000	4
En-Ta	197M	4096	0.0007	0.3	0.3	4000	4
Ru-En	276M	8192	0.0007	0.2	0.1	3584	128

Table 8: Baseline MT model hyper-parameters

	De-En		En-De		En-Ta		Ru-En	
TER	valid	test	valid	test	valid	test	valid	test
beam (fw)	60.9	58.0	67.1	63.2	85.1	88.2	52.7	52.3
+ MLM (Salazar et al., 2019)	60.9	58.2	66.4	62.6	85.7	88.8	52.5	52.2
+ MLM-ft (Salazar et al., 2019)	60.8	58.2	66.5	62.6	85.7	89.1	52.4	52.0
+ LM	59.7	57.1	65.8	61.6	84.8	88.6	51.9	51.3
NCD (Yee et al., 2019)	58.4	54.9	65.2	60.9	84.1	87.8	51.2	50.2
DrNMT	**57.7**	**54.1**	65.2	60.6	83.9	87.5	**50.5**	**49.3**
+ NCD	57.9	54.2	**64.9**	**60.1**	**83.5**	**87.4**	50.6	49.6

Table 9: Validation and test TER with beam size 50. The results for De-En and En-De are averaged from newstest2014 and 2015 for validation and newstest2016, 2017, 2018 and 2019 for test. The results for Ru-En are averaged from newstest2015 and 2016 for validation and newstest2017, 2018 and 2019 for test. DrNMT was trained using BLEU.

	valid			test				
TER	2014	2015	avg	2016	2017	2018	2019	avg
beam (fw)	61.8	59.9	60.9	56.1	60.0	53.4	62.4	58.0
+ MLM (Salazar et al., 2019)	61.7	60.0	60.9	56.1	60.1	53.5	63.2	58.2
+ MLM-ft (Salazar et al., 2019)	61.6	60.0	60.8	56.1	60.1	53.4	63.1	58.2
+ LM	60.7	58.7	59.7	54.9	59.0	52.4	61.9	57.1
NCD (Yee et al., 2019)	59.1	57.6	58.4	52.7	57.3	50.2	59.5	54.9
DrNMT	58.6	56.7	**57.7**	52.2	56.4	49.2	58.4	**54.1**
+ NCD	58.8	56.9	57.9	52.2	56.5	49.4	58.7	54.2

Table 10: Validation and test TER on WMT'19 De-En with beam size 50. DrNMT was trained using BLEU.

	valid			test				
TER	2014	2015	avg	2016	2017	2018	2019	avg
beam (fw)	68.5	65.7	67.1	61.8	67.7	56.0	67.3	63.2
+ MLM (Salazar et al., 2019)	67.7	65.0	66.4	61.0	66.9	55.4	67.1	62.6
+ MLM-ft (Salazar et al., 2019)	67.8	65.1	66.5	61.3	67.1	55.4	66.6	62.6
+ LM	67.0	64.6	65.8	60.5	66.2	54.6	65.1	61.6
NCD (Yee et al., 2019)	66.5	63.9	65.2	60	65.8	53.6	64.3	60.9
DrNMT	66.4	63.9	65.2	59.3	65.9	53.3	63.9	60.6
+ NCD	66.2	63.5	**64.9**	58.9	65.1	52.7	63.5	**60.1**

Table 11: Validation and test TER on WMT'19 En-De with beam size 50. DrNMT was trained using BLEU.

	valid			test			
TER	2015	2016	avg	2017	2018	2019	avg
beam (fw)	51.8	53.6	52.7	48.9	54.4	53.5	52.3
+ MLM (Salazar et al., 2019)	51.6	53.4	52.5	48.6	54.4	53.5	52.2
+ MLM-ft (Salazar et al., 2019)	51.5	53.3	52.4	48.4	54.4	53.2	52.0
+ LM	50.9	52.8	51.9	48.0	53.5	52.5	51.3
NCD (Yee et al., 2019)	50.3	52.1	51.2	47.1	52.5	51.1	50.2
DrNMT	49.6	51.3	**50.5**	46.2	52.1	49.7	**49.3**
+ NCD	49.7	51.5	50.6	46.4	52.1	50.4	49.6

Table 12: Validation and test TER on WMT'19 Ru-En with beam size 50. DrNMT was trained using BLEU.

	valid			test				
BLEU	2014	2015	avg	2016	2017	2018	2019	avg
beam (fw)	23.3	26.0	24.7	29.2	26.1	31.6	24.0	27.7
+ MLM (Salazar et al., 2019)	24.4	27.0	25.7	30.2	27.2	32.5	24.7	28.7
+ MLM-ft (Salazar et al., 2019)	24.4	27.1	25.8	30.3	27.3	32.7	24.9	28.8
+ LM	24.9	27.7	26.3	31.0	27.7	33.1	25.1	29.2
NCD (Yee et al., 2019)	26.0	28.4	27.2	32.8	29.0	34.9	26.7	30.9
DrNMT	26.2	28.9	27.6	33.2	29.6	35.6	27.5	31.5
+ NCD	26.6	29.1	**27.9**	33.5	29.9	35.9	27.8	**31.8**
oracle BLEU	31.8	34.7	33.3	39.2	35.2	41.6	33.7	37.4

Table 13: Validation and test BLEU on WMT'19 De-En with beam size 50.

	valid			test				
BLEU	2014	2015	avg	2016	2017	2018	2019	avg
beam (fw)	21.6	24.5	23.1	27.6	22.8	32.9	23.1	26.6
+ MLM (Salazar et al., 2019)	22.0	25.0	23.5	27.9	23.4	33.5	23.5	27.1
+ MLM-ft (Salazar et al., 2019)	22.3	25.1	23.7	28.4	23.4	34.0	24.3	27.5
+ LM	22.9	25.7	24.3	29.0	24.3	34.9	25.6	28.5
NCD (Yee et al., 2019)	23.3	26.3	24.8	29.7	24.6	35.7	26.4	29.1
DrNMT	23.2	26.2	24.7	29.9	24.3	35.4	26.3	29.0
+ NCD	23.6	26.6	**25.1**	30.4	25.1	36.3	27.0	**29.7**
oracle BLEU	29.6	33.1	31.4	37.1	31.2	43.6	31.6	35.9

Table 14: Validation and test BLEU on WMT'19 En-De with beam size 50.

	valid			test			
BLEU	2015	2016	avg	2017	2018	2019	avg
beam (fw)	33.3	33.6	33.5	36.8	32.3	33.9	34.3
+ MLM (Salazar et al., 2019)	33.8	33.8	33.8	37.2	32.7	34.5	34.8
+ MLM-ft (Salazar et al., 2019)	33.8	33.9	33.9	37.5	32.7	34.8	35.0
+ LM	34.5	34.6	34.6	38.1	33.7	35.6	35.8
NCD (Yee et al., 2019)	35.1	35.4	35.3	39.0	34.6	36.9	36.8
DrNMT	35.3	35.3	35.3	39.1	34.2	37.9	37.1
+ NCD	35.7	35.7	**35.7**	39.6	34.8	37.6	**37.3**
oracle BLEU	45.1	45.4	45.3	49.5	43.9	47.6	47.0

Table 15: Validation and test BLEU on WMT'19 Ru-En with beam size 50.

src: Zusammen waren wir ein unschlagbares Team.
ref: Together, we were an unbeatable team.
NCD: Together we were an impossible team.
DrNMT + NCD: Together we were an unbeatable team.
src: Keine neuen Flüchtlinge, das würde die Lage entspannen.
ref: The situation would ease a bit if they did not receive any new refugees.
NCD: No new refugees would ease the situation.
DrNMT + NCD: No new refugees, that would ease the situation.
src: Je mehr ich es ansehe, desto verwirrender wird es.
ref: The more I look at it, the more mind-boggling it becomes.
NCD: The more I say it, the more confusing it becomes.
DrNMT + NCD: The more I look at it, the more confusing it becomes.
src: Auf den Radarschirmen war die Form eines Dreamliners zu sein.
ref: The shape of a Dreamliner could be seen on radar screens.
NCD: The radar screens were the shape of a Dreamliner.
DrNMT + NCD: It was the shape of a Dreamliner on the radar screens.

Table 16: Examples where DrNMT + NCD is rated higher than NCD in human evaluation.

src: Nein, die ausgehandelten Software-Updates sind freiwillig.
ref: No, the brokered software updates are voluntary.
NCD: No, the negotiated software extension is voluntary.
DrNMT + NCD: No, the software process that is negotiated is voluntary.
src: Viele der attraktiveren (Hand-Desinfektionsmittel) sind diejenigen, die parfümiert sind.
ref: A lot of the more attractive (hand sanitizers) are the ones that are scented.
NCD: Many of the more attractive (hand disinfectant) tools are those that are coded.
DrNMT + NCD: Many of the more attractive (hand infections) are those that are coded.
src: Herr Schmidt, Wie kann sich der Verbraucher vor vergifteten Eiern schützen?
ref: Mr Schmidt, how can consumers protect themselves against poisoned eggs?
NCD: Mr Schmidt, how can consumers protect themselves from poisoned eggs?
DrNMT + NCD: Mr Schmidt, how can the consumer protect itself from poisoned eggs?
src: Sie benutzt sogar nur selten einen Topf.
ref: She rarely even uses a pot.
NCD: In fact, she rarely uses a pot.
DrNMT + NCD: It even rarely uses a pot.

Table 17: Examples where NCD is rated higher than DrNMT + NCD in human evaluation.