A Pre- and post-processing

We reuse the data processing of each pre-trained system (reusing subword segmentation models). For UEDIN models, the data is preprocessed using a SentencePiece (Kudo and Richardson, 2018) model with a joint vocabulary of 32k subwords. By default, we use a maximum sentence length of 100 subwords and scale this when adding previous context (e.g. 200 subwords for 1 previous sentence, 300 for 2, etc.). For FAIR models, the Moses toolkit (Koehn et al., 2007) is used for tokenisation and FastBPE [7] for subword segmentation (Sennrich et al., 2016b). A maximum length of 1024 is used for all models.

For FAIR models, we observed some inconsistencies while detokenising the generated outputs in terms of punctuation. We post-processed the output using regular expressions to ensure there was no additional space with the punctuation marks. We also standardised the production of $ in the German output such that all the prices now follow XX,XX $ convention.

B Hyper-parameters

The pretrained models are fine-tuned (first on filtered Paracrawl data, then on the task-specific training data). Adam optimiser (Kingma and Ba, 2015) is used to fine-tune all models, with a batch size of 32 (except for FAIR fine-tuning on filtered Paracrawl data where a batch size of 64 was used). For UEDIN, we use a learning rate of 0.0009, a learning rate warmup of 16000. We validate every 250k subwords decoded. The best model is chosen based on the best BLEU score and least cross-entropy loss on the side of the dev set specific to the language direction for UDEIN and FAIR respectively. For FAIR, we use a learning rate of the last epoch of the pre-trained model (9.85e-5 for en–de, 9.89e-5 for de–en) and validate per epoch.

The training parameters for each model are summarised in Table 6.

Detail\Model	UEDIN	FAIR
Preprocessing	SentencePiece[8]	Moses tokeniser[9]+ FastBPE[10]
Optimiser	Adam	Adam
Learning rate	9e-4 (warmup of 16000)	9.85e-5 (En-De), 9.89e-5 (De-En)
Batch size	32	32 (64 for paracrawl data)
Checkpoint	250k words decoded	1 training epoch
Best model	Best BLEU on dev	Smallest cross-entropy loss on dev

Table 6: Pre-processing and hyper-parameters.

[7] https://github.com/glample/fastBPE

[10] https://github.com/glample/fastBPE

[10] (Kudo and Richardson, 2018), using a joint 32k model.

[10] (Koehn et al., 2007)