		en-tr		tr-en	
Original copies	Paraphrase copies	Baseline	CT	Baseline	CT
1	0	13.3	-	15.0	-
10	1	12.4	**13.4**	15.6	**16.1**
5	1	11.9	13.1	15.1	16.0
1	1	10.7	12.8	13.5	**16.1**
0	1	10.1	12.6	13.2	15.9

Table 6: NMT augmentation results in BLEU. The first column (original copies) indicates how many times we repeat the original training corpus. Paraphrase copies indicates how many copies of the paraphrased corpus we include. Baseline systems are trained just on the indicated data, continued training (CT) systems are initialized with the baseline systems then continued trained on the original corpus.

A Alternative Paraphrase Generation for MNLI

In addition to the paraphrase generation for MNLI described in the main body of this paper (see Section 5.1), our algorithmic improvements to lexically-constrained decoding enabled us to try a more complicated paraphrase generation scheme for MNLI, but the model trained on the resulting augmented dataset did not improve upon the baseline. This alternative scheme had three components: **(1)** Given a premise-hypothesis pair, we identify the tokens common to both the premise and hypothesis and place a negative constraint on the top-IDF such token. If no common tokens exist, we consider the premise and hypothesis individually and place a negative constraint on their respective top-IDF token. **(2)** We place a positive constraint on gendered pronouns (e.g. "he", "she", etc.). **(3)** We positively constrain on automatically detected named entities.

The main difference between this alternative paraphrase generation scheme and the original scheme is that the alternative one uses positive constraints. We hypothesized that pronouns and named entities would be difficult for the rewriter to accurately paraphrase and wanted to see whether positively constraining on them would help maintain semantic similarity.

Training with this rewritten data did not lead to as large improvements on MNLI on the development set. Without ELMo (or aggregation), we achieve 74.3 F1, a 0.5 F1 drop from the baseline. With ELMo, we achieve 76.1 F1, a 0.3 F1 gain over the baseline but far from the 0.6 gain described in Section 5.1. Weighted aggregation on top of this did not help in either case; the best weights would ignore the rewritten sentence pairs. We did not further explore this paraphrase generation method.

B Tuning Weights for MNLI Aggregation

Since there are only four predictions for the ternary classification task, an exhaustive search is possible. The best weights are 3.1, 2, 2, and 1 for no rewritten, rewritten p, rewritten h and both rewritten examples respectively.

C Data Augmentation for Neural Machine Translation

We apply our paraphrastic rewriter to the task of machine translation to see if augmenting with paraphrases leads to improvements.

Data We use the parallel training corpus from the 2016 Conference on Machine Translation (WMT) shared task on Turkish-English Translation (Bojar et al., 2016b). The parallel training data consists of $207,373$ lines of news articles, while the development set consists of 1001 lines and the test set consists of 3000 lines (also news).

Paraphrase Generation For each English sentence in the training data, we generate paraphrases using the 34 heuristics described by Hu et al. (2019), which include both positive and negative constraints. Only unique paraphrases with a token Jaccard Index lower than 0.65 compared to the reference are kept. This lead to an average of 10 paraphrases per sentence. We then pair each paraphrase with the corresponding translation of the original English sentence to generate additional parallel text for machine translation training.

Model We train Neural Machine Translation systems using SOCKEYE. We build Transformer

models (Vaswani et al., 2017) with a 6-layer encoder and decoder with a model size of 512 and 8 attention heads. For the English text, we apply the preprocessing from Section 4. For the Turkish text, we tokenize and truecase the data using scripts from Moses (Koehn et al., 2007). We then learn a shared byte-pair encoding (BPE) over the entire WMT training data in both languages with 30,000 BPE operations (Sennrich et al., 2016). We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-4} and a learning rate reduce factor of .9. For continued training, we use a learning rate of 10^{-5}

Experimental Results We evaluate our models using BLEU score on `newstest2016`.

We compare a baseline system trained only on the WMT data with systems that include varying amounts of paraphrased data. The paraphrased text is approximately 10 times larger than the original data, so with 10 copies of the original data and one copy of the paraphrases, the data is about half from each source, but with the original data repeated. As shown in Table 6, for these systems (baseline columns), we see a .6 gain in English - Turkish translation when we sample the original data 10 times, but no improvement in Turkish - English Translation.

We also begin with each of the models of varying data amounts, and run continued training (CT) (Luong and Manning, 2015) with just the baseline data. This allows the models to have broader coverage from the paraphrased corpus in initial training, but to fine-tune on the original WMT training corpus. For these systems (CT columns), we see an improvement over the system that was used to initialize them. Additionally, when we continue train the model that was trained on 10 copies of the original corpus and one copy of that paraphrased corpus, we observe a 1.1 BLEU improvement in Turkish-English Translation over the system that was trained only on the original corpus.