JHU 2019 Robustness Task System Description

Matt Post and **Kevin Duh**
Johns Hopkins University
Baltimore, Maryland
USA

Abstract

We describe the JHU submissions to the French–English, Japanese–English, and English–Japanese Robustness Task at WMT 2019. Our goal was to evaluate the performance of baseline systems on both the official noisy test set as well as news data, in order to ensure that performance gains in the latter did not come at the expense of general-domain performance. To this end, we built straightforward 6-layer Transformer models and experimented with a handful of variables including subword processing (FR–EN) and a handful of hyperparameters settings (JA↔EN). As expected, our systems performed reasonably.

1 Introduction

The team at JHU submitted three systems to the WMT19 Robustness task: French–English, Japanese–English, and English–Japanese. Our goal was to evaluate the performance of reasonable state-of-the-art systems against both the robustness test set as well as more standard "general domain" test sets. We believe this is an important component of evaluating for actual robustness. In this way, we ensure that performance gains on robustness data are not purchased at the expense of this general-domain performance. Our systems used no monolingual data and relatively straightforward state-of-the-art techniques, and produced systems of roughly average performance.

2 French-English Systems

2.1 Training Data

We constrained our data use to the officially supplied data, comprising the WMT15 English–French parallel data (Bojar et al., 2015). For French, we experimented with three data settings:

- all of Europarl and News Commentary;
- the best million lines each of CommonCrawl, Gigaword, and the UN corpus; and
- the MTNT training data.

Data sizes are indicated in Table 1.

dataset	segments	words
Europarl	2.0m	50.2m
News Commentary	200k	4.4m
Common Crawl	820k	17.4m
FR–EN Gigaword	1m	26.1m
UN Doc	106k	1.1m
MTNT$_{EN \rightarrow FR}$	36k	841k
MTNT$_{FR \rightarrow EN}$	19k	634k

Table 1: Training datasets for French–English systems. Common Crawl, Gigaword, and the UN data are post-filtering.

To filter the data, we applied dual cross-entropy filtering (Junczys-Dowmunt, 2018). We trained two smaller 4-layer Transformer models, one each for EN–FR and FR–EN, and used them to score the data according to the formula:

$$\exp(-(|s_1 - s_2| + 0.5 * (s_1 + s_2)))$$

where s_1 is the score (a negative logprob) from the forward FR–EN model and s_2 the score from the reverse EN–FR model. We then uniqued this data, sorted by score, and took a random sample of one million lines from the set of all sentence pairs with a score greater than 0.1.[1] For all but FR–EN Gigaword, what remained was well less than a million lines. We did this both because prior work has indicated the utility of filtering, and to make our

[1] We determined this threshold by eyeballing where in the ranked list the garbage started to thin out.

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 552–558
Florence, Italy, August 1-2, 2019.

training data sizes more manageable. We therefore did not compare against a model trained on all of the filtered data.

We experimented with two preprocessing regimes. In the first, we applied standard preprocessing techniques from the Moses pipeline[2] (Koehn et al., 2007), followed by subword splitting with BPE (Sennrich et al., 2016) using 32k merge operations. In the second scenario, we did not use any data preparation, instead applying `sentencepiece` (Kudo and Richardson, 2018) with subword regularization (Kudo, 2018) directly to the raw text. In this latter setting, we varied the size of the learned subword models, experimenting with 8k, 16k, 24k, and 32k.

2.2 Models

We used Sockeye (Hieber et al., 2017), a sequence to sequence transduction framework written in Python and based on MXNet. Our models were variations of the Transformer architecture (Vaswani et al., 2017), mostly using default settings supplied with Sockeye: an embedding and model size of 512, a feed-forward layer size of 2048, 8 attention heads, and three-way tied embeddings. We used batch sizes of 4,096 words, checkpointed every 5,000 updates, and stopped training with the best-perplexity checkpoint when validation perplexity had failed to improve for 10 consecutive checkpoints. The initial learning rate was set to 0.0002, the Sockeye default.

2.3 Scoring

At test time, we decoded with beam search using a beam of size 12.

We scored with sacreBLEU (Post, 2018), with international tokenization.[3] In the spirit of the robustness task, we measure BLEU not just on the reddit dataset, but also on the WMT15 newstest dataset, in order to examine how experimental variables vary in both in- and out-of-domain settings. We believe that testing both in- and out-of-domain data is essential to measuring robustness.

2.4 Results & Discussion

Table 2 contains BLEU scores.

	WMT15	MTNT18
4 layers (BPE)	31.6	27.9
6 layers (BPE)	32.7	27.9
+ MTNT	32.6	32.9
+ filter	36.4	33.7
+ both	37.2	39.9
sp_{24k} + filter	36.5	34.5
sp_{24k} + both	37.2	40.0

Table 2: French–English translation results.

	WMT15		MTNT18	
size	filter	both	filter	both
8k	36.0	36.5	33.9	38.7
16k	36.2	36.9	33.9	39.7
24k	36.5	37.2	34.5	40.0

Table 3: BLEU scores with the sentencepiece models and no other preprocessing.

Observation 1 *Improvements are to be had both from more data and from better (in-domain) data.* Adding the large filtered dataset to the 6 layer model improved BLEU more (27.9 → 33.7, +5.8) than adding the MTNT training data (27.9 → 32.9, +5), but the gains from both were even greater (+12).

Observation 2 In order to ensure that our models did not increase accuracy on the MTNT data at the expense of in-domain data, we report scores on both WMT and MTNT test sets. In only one situation was there a problem: For the 6-layer Transformer, adding the MTNT data alone (without the large amount of filtered bitext) helped on MTNT$_1$8 (+5) but caused a small drop on WMT15 (-0.1).

Observation 3 In all situations, the sentencepiece model (with no other preprocessing) was just as good as the BPE model (with the Moses preprocessing pipeline). In one situation (adding the filtered data alone), it caused a gain of 0.8 over its BPE counterpart.

We further conducted a small experiment varying the sentencepiece model size (Table 3). Larger sentencepiece models were consistently better in this relatively large-data setting.

Our score on the official MTNT2019 blind test set was 40.2.

[2] Calling `normalize-punctuation.perl`, `remove-non-printing-char.perl`, and `tokenizer.perl` with flags `-no-escape`.

[3] BLEU+case.mixed+refs.1+smooth.exp+tok.intl+v1.2.20

3 Japanese-English Systems

3.1 Training Data

We trained systems using only the bitext data allowed in the shared task constrained setting:

- The in-domain Reddit dataset–MTNT version 1.1 (Michel and Neubig, 2018)[4]–consists of approximately 6k segments for training (which we label Train-MTNT) and 900 segments for validation (Valid-MTNT) in both JA→EN and EN→JA language directions. Additionally we use the included ”test set” (which we label Test18-MTNT) for internal BLEU benchmarks prior to submitting results for the official 2019 blindtest. We did not use the monolingual part of MTNT.

- The out-of-domain data consists of KFTT (Wikipedia articles), TED Talks, and JESC Subtitles.[5] We concatenate these out-of-domain training data with Train-MTNT to create Train-ALL; similarly we concatenate the out-of-domain validation data with Valid-MTNT to create Valid-ALL.

Dataset sizes are shown in Table 4.

JA→EN dataset	segments	words
Train-ALL	3.9m	42.7m
Train-MTNT	6506	155k
Valid-ALL	5416	88k
Valid-MTNT	965	23k
Test18-MTNT	1001	13k
EN→JA dataset	**segments**	**words**
Train-ALL	3.9m	42.9m
Train-MTNT	5775	333k
Valid-ALL	5405	111k
Valid-MTNT	954	46k
Test18-MTNT	1002	13k

Table 4: Datasets for English–Japanese systems. Word counts are source side only.

For preprocessing on the English side, we apply the standard Moses pipeline in the same fashion as the French–English system. For preprocessing on the Japanese side, we first performed word segmentation by Kytea (Neubig et al., 2011)[6], then ran the English Moses preprocessing pipeline to handle potential code-switched English/Japanese in the data. Finally, we induced BPE subword units with 10k, 30k, and 50k merge operations, independently for each side on the bitexts (JA→EN Train-ALL and EN→JA Train-ALL). Unlike the French-English systems, the Japanese-English systems do not use shared BPE and embeddings.

3.2 Models

We use the Sockeye Transformer models for both JA→EN and EN→JA directions, similar to our French-English systems. The hyperparameter settings are different, however. We performed random search in the following hyperparameter space (see Table 5):

- Initial learning rate (**LR**) for the ADAM optimizer: 0.001, 0.0003, 0.0006

- Number of attention heads (**head**): 8, 16

- Number of layers (**layer**): 2, 4

- Feed-forward layer size (**ffsize**): 1024, 2048

- Embedding and model size (**embedding**): 256, 518, 1024

The training process follows a continued-training procedure (c.f. Koehn et al. (2018); Khayrallah et al. (2018)): In Stage 1, we train systems from scratch on Train-ALL, and perform early stopping on Valid-ALL. This represents a mixed corpus with both in-domain and out-of-domain bitexts. For all models, we used batch sizes of 4,096 words, checkpointed every 2,000 updates, and stopped training with the best-perplexity checkpoint when validation perplexity on Valid-ALL had failed to improve for 16 consecutive checkpoints.

In Stage 2, we fine-tuned the above systems by training on Train-MTNT, and perform early stopping on Valid-MTNT. Effectively, we initialize a new model with Stage 1 model weights, reset the optimizer's learning rate schedule, and train on only in-domain data. To prevent overfitting to the small Train-MTNT bitext, we now checkpoint

[4] http://www.cs.cmu.edu/~pmichel1/mtnt/

[5] The data is also downloaded in pre-packaged form from the MTNT website via https://github.com/pmichel31415/mtnt/releases/download/v1.1/clean-data-en-ja.tar.gz, but do not confuse these with the MTNT data, which is in the Reddit domain.

[6] v0.4.7: http://www.phontron.com/kytea/

more frequently, saving a checkpoint after every 50 updates, and stopped training either when the perplexity on Valid-MTNT fails to improve for 16 consecutive checkpoints or when we reached 30 checkpoints (i.e., 30 × 50 = 1500 updates of 4,096 word batches), to prevent fitting excessively on the Train-MTNT bitext.

3.3 Scoring

At test time, we decoded with beam search using a beam of size 5. We scored with sacreBLEU (Post, 2018), with international tokenization.[7] Per organizer suggestion, we applied Kytea to Japanese output prior to scoring. We measure BLEU on both VALID-ALL and Test18-MTNT in order to compare the results on mixed and in-domain corpora.

3.4 Results & Discussion

The BLEU results for Stage 1 models are shown in Table 5. We performed random search in hyperparameter space, training approximately 40 models in each language-pair. The table is sorted by Test18-MTNT BLEU score and shows the top 5 models in terms of BLEU (id=a,b,c,d,e; id=z,y,x,w,v) as well as another 5 randomly selected model (id=e,f,g,h,i,j; id=u,t,s,r,q).

Observation 1: Despite the relatively narrow range of hyperparameter settings, there is a comparatively large range of BLEU scores in the table. For example in JA→EN, the best Test18-MTNT BLEU is 11.1, 2.7 points better than the worst BLEU (8.4) in the table; there are other poorer performing systems, not sampled for the table. This suggests that hyperparameter search is important in practice, even for relatively standard hyperparameters.

Additionally, we note it is difficult to make posthoc recommendations on the "best" hyperparameter settings, as there are no clear trends in the data. For example, from the top 5 JA→EN models, it appears that 30k BPE merge operations is good, but there is an competitive outlier with 10k BPE (id=c). In the results (not all shown in the table), most 10k BPE models achieve Test18-MTNT BLEU in the 8-10 range, so it is difficult to explain the strong BLEU score of id=c. Also, it does appear that layer=4 is consistently better than layer=2 in the JA→EN results, but the results are more mixed in the EN→JA direction.

Observation 2: There is some correlation between the BLEU scores of Valid-ALL and Test18-MTNT; the system rankings are relatively similar. But we note that there are a few outliers, e.g. the top 5 models in EN→JA perform similarly on Test18-MTNT, but there are noticeable degradations for id=x and id=v on Valid-ALL. Similarly, id=b and id=c perform close on Test18-MTNT but not on Valid-ALL. With the goal of robustness, we think these kinds of BLEU gaps due to domain differences deserve more investigation.

Continued Training: Next, we perform continued training on the top 5 models. The results on Test18-MTNT are shown in Table 6. We observe consistent BLEU gains in these Stage 2 models, close to 2 or 3 points across all systems. This re-affirms the surprising effectiveness of a simple procedure such as continued training; but we should also note that preliminary efforts on English-French did not yield similar gains.

Note that we do not measure Valid-ALL in this case since we now expect the models to be optimized specifically for MTNT; it is likely Valid-ALL scores will degrade due to catastrophic forgetting (Thompson et al., 2019).

Final Submission: In the final official submission, we performed an 4-ensemble of the Stage 2 Continued Training models of id=a,b,d,e for JA→EN and id=z,y,w,v for EN→JA. Note that the ensemble method in Sockeye currently assumes the same vocabulary, so BPE needs to be the same for all models in the ensemble. This is a reasonable assumption, but in the spirit of subword regularization (Kudo, 2018), we think it may be interesting to explore whether ensembles of systems with diverse BPE will lead to more robust outputs.

For JA→EN, the ensemble achieved 14.6 BLEU on Test18-MTNT (N-gram precisions: 43.9/19.3/10.1/5.5, Brevity Penality: 0.991, Length ratio: 0.991). The official MTNT2019 blindtest cased-BLEU is 11.4.

For EN→JA, the ensemble achieved 15.0 BLEU on Test18-MTNT (N-gram precisions: 45.2/19.2/10.3/5.7, Brevity Penality: 1.0, Length ratio: 1.122). The official MTNT2019 blindtest case-BLEU is 14.8.

4 Conclusion

We constructed reasonably-scoring systems on three language pairs without too much effort. Our

[7] BLEU+case.mixed+refs.1+smooth.exp+tok.intl+v1.2.14

	JA→EN Systems Hyperparameter Setting						BLEU (EN output)	
id	**BPE**	**LR**	**head**	**layer**	**ffsize**	**embed**	**Valid-ALL**	**Test18-MTNT**
a	30k	0.0003	8	4	2048	512	17.1	11.1
b	30k	0.0006	16	4	2048	512	16.5	10.7
c	10k	0.0006	16	4	2048	512	15.7	10.5
d	30k	0.0006	16	4	2048	256	16.4	10.1
e	30k	0.0003	8	4	1024	256	16.0	10.0
f	50k	0.0003	8	4	1024	512	16.4	10.0
g	30k	0.0006	8	2	2048	512	15.9	9.9
h	50k	0.0006	8	2	1024	256	14.4	9.1
i	10k	0.0006	8	2	2048	256	14.0	8.6
j	30k	0.0006	16	2	1024	1024	13.9	8.4

	EN→JA Systems Hyperparameter Setting						BLEU (JA output)	
id	**BPE**	**LR**	**head**	**layer**	**ffsize**	**embed**	**Valid-ALL**	**Test18-MTNT**
z	50k	0.0006	8	4	2048	256	17.0	12.7
y	50k	0.0003	16	4	2048	512	17.5	12.7
x	30k	0.0003	8	2	2048	512	16.6	12.6
w	50k	0.0006	16	4	2048	512	17.1	12.5
v	50k	0.001	8	4	2048	512	16.5	12.5
u	10k	0.0003	8	4	1024	512	16.4	12.3
t	30k	0.001	16	4	1024	256	16.0	12.1
s	50k	0.001	8	4	1024	256	15.8	12.1
r	10k	0.0006	16	2	1024	512	15.3	11.9
q	10k	0.0006	8	2	1024	256	14.5	10.6

Table 5: JA→EN and EN→JA Results for Stage 1 models. For each language pair, we show the top 5 models (according to Test18-MTNT) and another random selection of 5 models from randomized hyperparameter search.

id	Stage 1	Stage 2	Improvement
	JA→EN		
a	11.1	13.4	+2.3
b	10.7	13.4	+2.7
c	10.5	13.1	+2.6
d	10.1	13.1	+3.0
e	10.0	13.2	+3.2
	EN→JA		
z	12.7	14.5	+1.8
y	12.7	14.4	+1.7
x	12.6	14.5	+1.9
w	12.5	14.4	+1.9
v	12.5	14.3	+1.8

Table 6: Continued Training BLEU results on Test18-MTNT. Stage 1 results are from Table 5. Continued Training (Stage 2) consistently improves BLEU.

scores fell into roughly the middle tier among those reported on `matrix.statmt.org`. It is certain that much higher gains could be had by adding even known techniques to our pipeline, such as backtranslating monolingual data (Sennrich et al., 2016).

We also believe that our approach of evaluating on multiple test sets is essential to the robustness task. Without this, the task reduces to domain adaptation, and one has no assurance that high scores on the out-of-domain data do not come at the expense of general-domain performance.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44. Association for Computational Linguistics.

Philipp Koehn, Kevin Duh, and Brian Thompson. 2018. The jhu machine translation systems for wmt 2018. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 442–448, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, 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, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Brian Thompson, Huda Khayrallah Jeremy Gwinnup, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics.

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