Findings of the Fourth Workshop on Neural Generation and Translation

Kenneth Heafield[♠], **Hiroaki Hayashi**[♢], **Yusuke Oda**[♣], **Ioannis Konstas**[△], **Andrew Finch**[♡], **Graham Neubig**[♢], **Xian Li**[⋆], **Alexandra Birch**[♠]

[♢]Carnegie Mellon University, [♣]Google Research, [♠]University of Edinburgh
[△]Heriot-Watt University, [♡]Apple, [⋆]Facebook

Abstract

We describe the finding of the Fourth Workshop on Neural Generation and Translation, held in concert with the annual conference of the Association for Computational Linguistics (ACL 2020). First, we summarize the research trends of papers presented in the proceedings. Second, we describe the results of the three shared tasks 1) efficient neural machine translation (NMT) where participants were tasked with creating NMT systems that are both accurate and efficient, and 2) document-level generation and translation (DGT) where participants were tasked with developing systems that generate summaries from structured data, potentially with assistance from text in another language and 3) STAPLE task: creation of as many possible translations of a given input text. This last shared task was organised by Duolingo.

1 Introduction

Neural sequence to sequence models (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) are the workhorse behind a wide variety of different natural language processing tasks such as machine translation, generation, summarization and simplification. The 4th Workshop on Neural Machine Translation and Generation (WNGT 2020) provided a forum for research in applications of neural models to machine translation and other language generation tasks (including summarization, NLG from structured data, dialog response generation, among others). Overall, the workshop was held with two goals. First, it aimed to synthesize the current state of knowledge in neural machine translation and generation: this year we continued to encourage submissions that not only advance the state of the art through algorithmic advances, but also analyze and understand the current state of the art, pointing to future research directions. Towards this goal, we received a number of high-quality research contributions on the workshop topics, as summarized in Section 2. Second, the workshop aimed to expand the research horizons in NMT: we continued to organize the Efficient NMT task which encouraged participants to develop not only accurate but computationally efficient systems. This task had three participants each with a number of individual systems. We organized the second shared task on "Document-level Generation and Translation", which aims to push forward document-level generation technology and contrast the methods for different types of inputs. Unfortunately this task only had one participant. Finally, we introduced a new shared task, organised by Duolingo, which encouraged models to produce as many correct translations as possible for a given input. This task generated a lot of interest and there were 11 participants. The results of the shared task are summarized in Sections 3, 4 and 5.

2 Summary of Research Contributions

Similar to last year we invited the MT and NLG community to contribute to the workshop with long papers, extended abstracts for preliminary work, and cross-submissions of papers that have appeared in other venues. Keeping up with with the main vision of the workshop, we were aiming for a variety of works at the intersection of Machine Translation and Language Generation tasks.

We received a total of 28 submissions, from which we accepted 16. There were 2 cross-submissions, 3 extended abstracts and 11 full papers. There were also 15 system submission papers. We elicted two double-blind reviews for each submission, avoiding conflicts of interest.

With regards to thematology there were 8 papers with a focus on Natural Language Generation and 8 with the application of Machine Translation

Proceedings of the 4th Workshop on Neural Generation and Translation (WNGT 2020), pages 1–9

www.aclweb.org/anthology/D19-56%2d

in mind. The underlying emphasis across submissions was placed this year on capitalizing on the use of pre-training models (e.g., BERT; (Devlin et al., 2019) especially for low-resource datasets. The quality of the accepted publications was very high; there was a significant drop in numbers though in comparison to last year (36 accepted papers from 68 submissions) which is most likely due to the extra overhead on conducting research under lockdown policies sanctioned globally due to COVID-19 pandemic.

3 Efficiency Task

The efficiency task complements machine translation quality evaluation campaigns by also measuring and optimizing the computational cost of inference. This is the third edition of the task, updating and building upon the second edition of the task (Hayashi et al., 2019).

We asked participants to build English→German machine translation systems following the data condition of the 2019 Workshop on Machine Translation (Barrault et al., 2019) and submit them as Docker containers. Docker contains enabled consistent measurement of computational cost on several dimensions: time, memory, and disk space. These are measured under three hardware conditions: a GPU, a single CPU core, and multi-core CPU on all cores. Participants were free to choose what metrics and hardware platforms to optimize for.

Three teams submitted to the shared task: NiuTrans, OpenNMT, and UEdin. All teams submitted to the GPU and multi-core CPU tracks; OpenNMT and UEdin submitted to the single-CPU track. Some CPU submissions from UEdin had a memory leak; their post-deadline fix is shown as "UEdin Fix."

Common techniques across teams were variations on the transformer architecture, model distillation, 16-bit floating point inference on GPUs (except OpenNMT), and 8-bit integer inference on CPUs (except NiuTrans). Curiously, all submissions used autoregressive models despite the existence of non-autoregressive models motivated by speed.

3.1 Hardware

The GPU track used a `g4dn.xlarge` instance with one NVIDIA T4 GPU, 16 GB GPU RAM, 16 GB host RAM, and 2 physical cores of an Intel Xeon Platinum 8259CL CPU. The NVIDIA T4 GPU is relatively small compared to the NVIDIA V100 GPU, but the newer Turing architecture introduces support for 4-bit and 8-bit integer operations in Tensor Cores. In practice, however, participants used floating-point operations on the GPU even though both OpenNMT and UEdin used 8-bit integers in their CPU submissions. This was primarily due to code readiness. Timing was run on a non-exclusive virtual machine because the instance is not yet available without virtualization.

The CPU tracks used a `c5.metal` instance which has two sockets of the Intel Xeon Platinum 8275CL CPU, 48 physical cores, hyperthreading enabled, and 192 GB RAM. As a Cascade Lake processor, it supports the Vector Neural Network Instructions (VNNI) that OpenNMT and UEdin used for 8-bit integer matrix multiplication. For the single core track, we reserved the entire machine then ran Docker with `--cpuset-cpus=0`. For the multi-core track, participants were free to configure their own CPU sets and affinities. The `c5.metal` instance runs directly on the full hardware; it is not a virtual machine.

Teams were offered AWS time to tune their submissions on the test hardware. All participants experimented on the test hardware using provided time or their own funds.

3.2 Measurement

Previous editions of the task specified the test set, but last year's organizers removed a team for generating the test outputs even with empty input. Moreover, translation time for some submissions was approaching one second and often lower than loading time. Hence we updated the task to make it more robust to adversarial participants while also increasing reliability of speed measurements. We told participants the test set would have one million lines, lines would have at most 100 space-separated words, source sentences from an unspecified quality evaluation corpus would be hidden in their input, and quality would be evaluated with BLEU.

After the submission deadline, we announced the main quality score is the unweighted average SacreBLEU[1] (Post, 2018) on WMT test sets from 2010–2019, excluding 2012.[2] The 2012 test set

[1] `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+test.wmt*+tok.13a+version.1.4.8` for various WMT test sets

[2] Participants are likely to have used these test sets in development. The WMT 2020 test set was not yet available and others were out of the domain the systems were trained for.

Corpus	Lines	Words	Characters
EMEA	759876	13152485	86584513
Tatoeba	214943	1398154	7303297
Federal	785	13458	87724
WMT10	2489	54021	328648
WMT11	3003	65829	396884
WMT13	3000	56089	332972
WMT14	2737	54268	329121
WMT15	2169	40771	241016
WMT16	2999	56789	337711
WMT17	3004	56435	336817
WMT18	2998	58628	351779
WMT19	1997	42034	249742
Total	1000000	15048961	96880224

Table 1: Size of corpora in the efficiency task input.

was excluded because it has lines longer than 100 words. We refer to this score as WMT1* while also reporting the usual WMT19 scores for the translation task.

Shown in Table 1, the test set consisted of the aforementioned WMT input sentences and filler. For filler, we used parallel corpora outside the WMT data condition to verify that the system was still translating reasonably. Specifically, we used a recent crawl of the European Medicines Agency (EMEA),[3] the Tateoba project,[4] and a crawl of the German Federal Foreign Office Berlin[5] all gathered by the European Language Resource Consortium. We do not consider the filler corpora clean or in-domain enough to be official evaluations of quality; results appear in supplementary material. To meet our promise to participants that lines would not be longer than 100 words (space-separated tokens), we excluded WMT12 and removed any English sentences longer than 100 words from the filler. We then truncated the German Federal Foreign Office Berlin corpus to obtain a total of 1 million lines. The input sentences were randomly shuffled and mixed across corpora, retaining a separate file to enable reconstruction. The final corpus and evaluation tools are available at `http://data.statmt.org/heafield/wngt20/test/`.

Time was measured with wall (real) time reported by `time` and CPU time reported by the kernel for the process group. We no longer measure loading time because it is small compared to the cost of translating 1 million sentences, is easy to game with busywork, and some toolkits do lazy initialization which makes loading time difficult to measure.

Peak RAM consumption was measured using `memory.max_usage_in_bytes` from the kernel for the CPU and by polling `nvidia-smi` for the GPU. Swap was disabled.

Participants were told to separate their Docker images into model and code files so that models could be measured separately from the relatively noisy size of code and libraries. A model was defined as "everything derived from data: all model parameters, vocabulary files, BPE configuration if applicable, quantization parameters or lookup tables where applicable, and hyperparameters like embedding sizes." Code could include "simple rule-based tokenizer scripts and hard-coded model structure that could plausibly be used for another language pair." They were also permitted to use standard compression tools such as `xz` to compress models; decompression time was included in results but small relative to the cost of translation. We report size of the model directory and Docker image size, both captured before the model ran.

Each evaluation started from a fresh boot of a constant Ubuntu 18.04 LTS disk image (one for CPU and one for GPU). Internet access was blocked at the cloud provider level except for the evaluation controller. This also prevented automatic upgrades.

3.3 Results

Measurements are reported in Table 2. The trade-offs between quality, model size, speed, and RAM are shown in Figure 1. We compare the cost-effectiveness of GPU and multi-core CPU hardware at the prices charged by Amazon Web Services in Figure 2.

Every team had a Pareto optimal submission for speed. This is largely due to teams focusing on different parts of the Pareto curve. OpenNMT focused on fast, small, and lower-quality systems plus one higher-quality submission. UEdin focused on higher-quality systems that were slower. Two of NiuTrans's four GPU submissions were Pareto optimal on speed, lying between OpenNMT and UEdin; their multi-core CPU submission performed poorly on all metrics.

Regarding model size, OpenNMT and UEdin made a range of Pareto-optimal submissions,

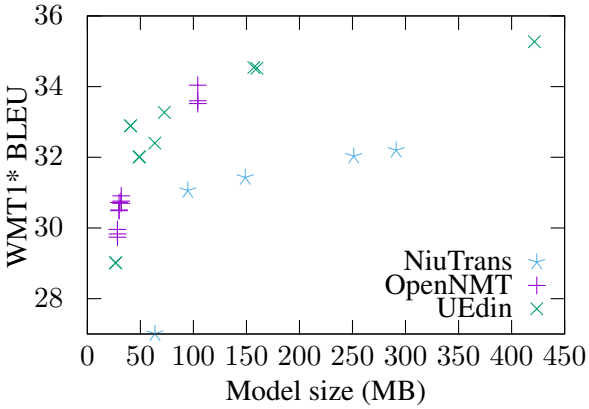
[3] https://edin.ac/2TSPnC7
[4] https://edin.ac/2ywYp01
[5] https://edin.ac/3bWrBes

NVIDIA T4 GPU									
		BLEU		Seconds		Disk MB		RAM MB	
Team	Variant	WMT19	WMT1*	Wall	CPU	Model	Docker	CPU	GPU
UEdin	large	42.9	35.3	5441	5462	422	933	5463	4992
UEdin	base	42.7	34.5	2385	2406	157	668	3793	3196
OpenNMT	base	42.9	34.0	2328	2377	104	308	488	1528
UEdin	tiny.untied	41.9	33.3	1971	1994	73	584	3146	2514
UEdin	tiny.push.i6	41.1	32.4	1536	1558	64	579	1000	1228
NiuTrans	35_6	40.9	32.2	3166	3450	291	887	2115	7748
NiuTrans	35_1	40.7	32.0	2023	2318	251	847	2115	5700
NiuTrans	18_1	40.2	31.4	1355	1646	149	745	2117	5700
NiuTrans	9_1	40.0	31.1	978	1260	95	691	2117	5444
OpenNMT	4-3-256-2ffn	40.0	30.9	762	812	32	235	388	1256
OpenNMT	6-3-256	39.9	30.7	731	782	30	233	393	892
OpenNMT	4-3-256	38.9	30.0	706	758	28	232	402	1064

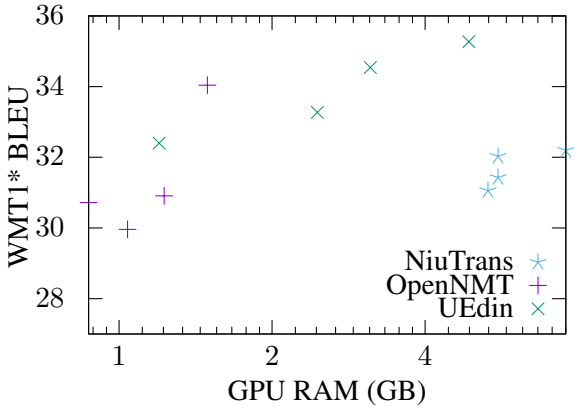
Single core Intel Cascade Lake CPU								
		BLEU		Seconds		Disk MB		RAM MB
Team	Variant	WMT19	WMT1*	Wall	CPU	Model	Docker	CPU
UEdin	base32	42.6	34.5	18649	18648	160	659	1728
UEdin Fix	base8	42.5	34.3	9128	9127	54	751	2001
OpenNMT	base	42.2	33.6	15978	15977	104	198	378
UEdin	tiny	41.6	32.9	14634	14634	41	737	164686
UEdin Fix	tiny	41.6	32.9	4799	4799	34	559	1549
UEdin	tiny.steady.i12	40.8	32.0	14553	14553	49	578	163388
UEdin	tiny.pushy.i6	40.5	32.0	14399	14399	49	578	164427
UEdin Fix	tiny.steady.i12	40.8	32.0	4577	4577	49	587	674
UEdin Fix	tiny.pushy.i6	40.5	32.0	4554	4554	49	587	675
OpenNMT	4-3-256-2ffn	39.8	30.8	3922	3922	32	125	238
OpenNMT	6-3-256	39.5	30.5	3717	3717	30	123	233
OpenNMT	4-3-256	38.7	29.8	3348	3348	28	122	220
UEdin	micro.voc8k	37.5	29.0	7184	7184	27	723	77158
UEdin Fix	micro.voc8k	37.5	29.0	4660	4660	19	716	2540

Multi-core Intel Cascade Lake CPU								
		BLEU		Seconds		Disk MB		RAM MB
Team	Variant	WMT19	WMT1*	Wall	CPU	Model	Docker	CPU
OpenNMT	base	42.0	33.5	795	38300	104	198	1552
UEdin	tiny	41.5	32.9	215	10014	41	737	108124
UEdin Fix	tiny	41.5	32.9	210	9840	34	737	28890
OpenNMT	4-3-256-2ffn	39.7	30.7	181	8735	32	125	1283
OpenNMT	6-3-256	39.4	30.5	155	7471	30	123	904
OpenNMT	4-3-256	38.6	29.7	144	6959	28	122	958
UEdin	micro.voc8k	37.4	29.0	188	8711	27	723	77157
UEdin Fix	micro.voc8k	37.4	29.0	190	8768	19	723	35051
NiuTrans	cpu	33.8	27.0	811	36198	64	432	19732

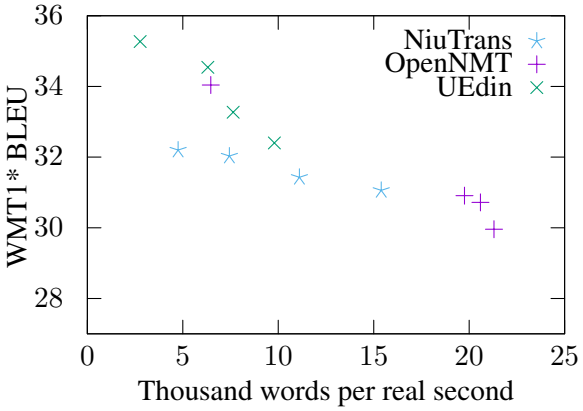
Table 2: Submissions to the efficiency shared task sorted in decreasing order of WMT1* BLEU. Systems translated 1,000,000 lines with 15,048,961 space-separated words.

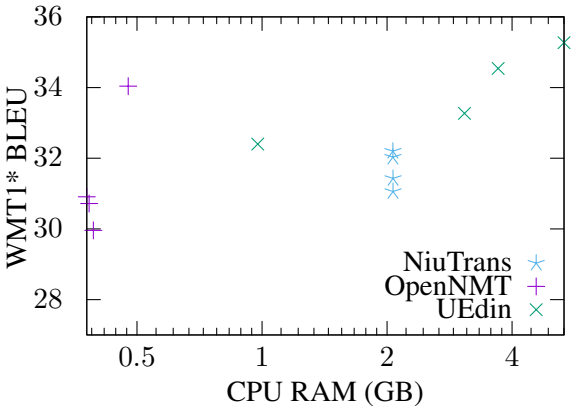


(a) Model size on disk regardless of hardware.

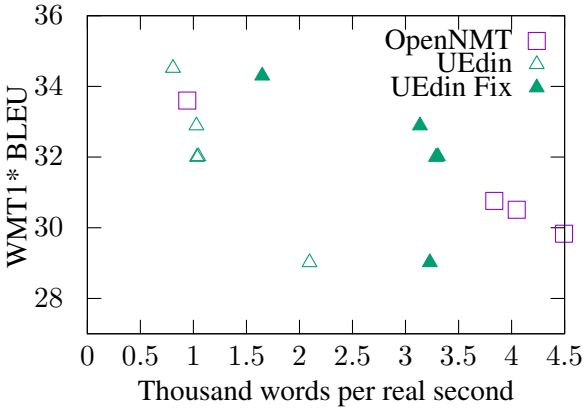


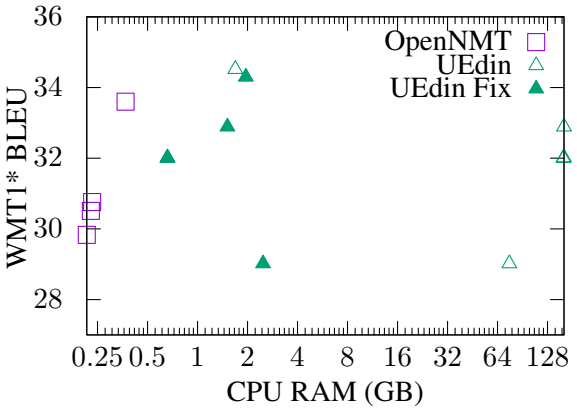
(b) Peak GPU RAM usage.



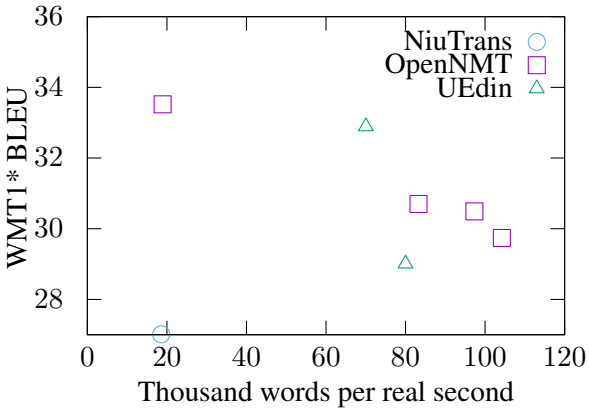


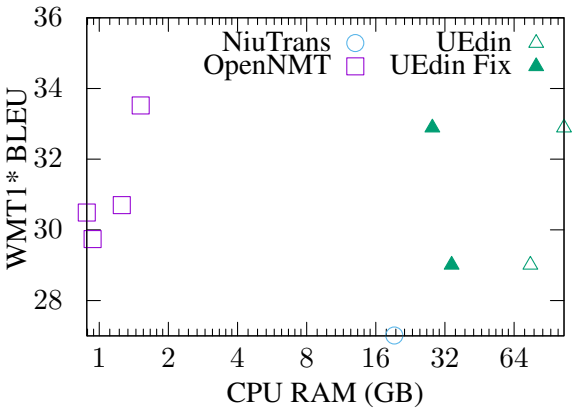
(c) GPU submissions including host CPU memory usage. GPU RAM is shown above.





(d) Single core CPU submissions.





(e) Multi-core CPU submissions; UEdin's fixed submissions had similar speed.

Figure 1: Performance of Efficiency Task Submissions.

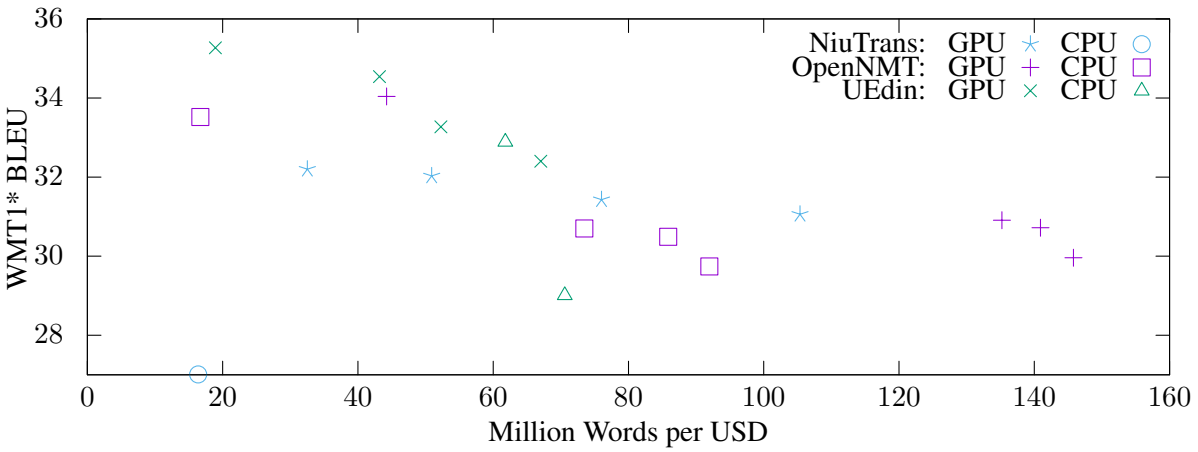


Figure 2: Price comparison of GPU and multi-core CPU submissions based on Amazon Web Services pricing of $4.08/hr for the `c5.metal` CPU instance and $0.526/hr for a `g4dn.xlarge` GPU instance. A single CPU core does not have a well-defined price.

mostly driven by the number of parameters and 8-bit quantization.

OpenNMT's small lower-quality models have low CPU RAM and Docker image size; UEdin is Pareto-optimal for higher-quality models. OpenNMT was the only team to optimize for these metrics in their system description. In their multi-core CPU submission, OpenNMT shared memory amongst processes while other participants simply used multiple processes with copies of the model.

4 Document Generation and Translation Task

Following the previous workshop, we continued with the shared task of document-level generation and translation. This task is motivated as the central evaluation testbed for document-level generation systems with different types of inputs by providing parallel dataset consisting of structured tables and text in two languages. We host various tracks within the testbed based on input and output constraints and investigate and contrast the system differences.

In particular, we conducted the following six tracks:

- **NLG (Data → En, Data → De):** Generate a document summary in the target language given only structured tables (*i.e.*, data-to-text).
- **MT (De ↔ En):** Translate a document in the source language to the target language (*i.e.*, document-level translation).
- **MT+NLG (Data+En → De, Data+De → En):** Generate a document summary given the structured tables and the summary in another language.

4.1 Evaluation Measures

We employ standard evaluation metrics for the tasks above along two axes following (Hayashi et al., 2019):

Textual Accuracy: BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) as measures for surface-level texutal accuracy compared to reference summaries.

Content Accuracy: Relation generation (RG), content selection (CS), and content ordering (CO) metrics (Wiseman et al., 2017) to assess the fidelity of the *content* to the input data.

An information extraction model is employed for content accuracy measures for each target language. We followed (Wiseman et al., 2017) and ensembled six information extraction models (three CNN-based, three LSTM-based) with different random seeds.

4.2 Data

We re-use Rotowire English-German dataset (Hayashi et al., 2019), which consists of a subset of the Rotowire dataset (Wiseman et al., 2017) with professional German translations. Each instance corresponds to an NBA game and consists of a box-score table for the match, base

information about the teams (*e.g.* team name, city), English game summary, and the same game summary translated to German. Final evaluation was performed on the test split of the Rotowire English-German dataset.

We followed the same setting in terms of additional resources participants could adopt.

Systems conforming to the data requirements are marked constrained, otherwise unconstrained. Results are indicated by the initials (C/U).

4.3 Baselines

We prepared two baselines for different tracks:

FairSeq-19 We use FairSeq (Ng et al., 2019) (WMT'19 single model[6]) for MT and MT+NLG tracks.

NCP+CC: We use a two-stage model from (Puduppully et al., 2019) for NLG tracks. English model was with the pretrained weights by the author and German model was trained only on Rotowire English-German dataset.

4.4 Submitted Systems

One team participated in the task, who focused on the German-English MT track of the task.

Team FJWU developed a system around Transformer-based sequence-to-sequence model. Additionally, the model employed hierarchical attention following (Miculicich et al., 2018) for both encoder and decoder to account for the document-level context. The system was trained in a two-stage process, where a base (sentence-level) NMT model was trained followed by the training of hierarchcal attention networks component. To handle the scarcity of in-domain translation data, they experimented with upsizing the in-domain data up to three times to construct training data. Their ablation experiments showed that this upsizing of in-domain data is effective at increasing the BLEU score.

4.5 Results

We show the MT track results in Table 3. We confirm that the use of both document-level models and in-domain data helps achieve better BLEU score, which has also been shown from the last workshop (Hayashi et al., 2019).

[6]Model identifier: `transformer.wmt19.en-de`, `transformer.wmt19.de-en`.

System	BLEU	Type
FJWU	45.04	C
FairSeq-19	42.91	C

Table 3: DGT results on the MT track (De → En).

5 STAPLE Task

Machine translation systems are typically trained to produce a single output, but in certain cases, it is desirable to have many possible translations of a given input text. At Duolingo, the world's largest online language-learning platform,[7] we grade translation-based challenges with sets of human-curated acceptable translation options. Given the many ways of expressing a piece of text, these sets are slow to create, and may be incomplete. This process is ripe for improvement with the aid of rich multi-output translation and paraphrase systems. To this end, we introduce a shared task called STAPLE: Simultaneous Translation and Paraphrasing for Language Education (Mayhew et al., 2020).

5.1 Task Description

In this shared task, participants are given a training set consisting of 2500 to 4000 English sentences (or *prompts*), each of which is paired with a list of comprehensive translations in the target language, weighted and ordered by normalized learner response frequency. At test time, participants are given 500 English prompts, and are required to produce the set of comprehensive translations for each prompt. We also provide a high-quality automatic reference translation for each prompt, in the event that a participant wants to work on paraphrase-only approaches. The target languages were Hungarian, Japanese, Korean, Portuguese, and Vietnamese.

5.2 Submitted Systems

There were 20 participants who submitted to the development phase, 14 participants who submitted to the test phase, and 11 participants who submitted system description papers. Submission models largely consisted of high-quality machine translation systems fine-tuned on in-domain shared task data from Duolingo, with different tricks for training, ensembling, and output filtering.

In the test phase, three teams submitted to all 5 language tracks, and one team submitted to two

[7]www.duolingo.com

tracks (Portuguese, and Hungarian). Of the remaining single-language submissions, Portuguese and Japanese were the most popular. In these single language submissions, teams did not tend to take language-specific approaches.

5.3 Results

Submission performance varied widely, but nearly all submissions improved significantly over organizer-provided baselines. The top submissions have comparable scores to taking the top 5 translations from each gold translation set.

Techniques popular among the more successful teams included weighting of training data according to learner response frequency, and classifier-based output filtering. Interestingly, techniques such as diverse beam search and beam reranking did not appear to improve results, despite their close relevance to the task. For more details and analysis, see Mayhew et al. (2020).

6 Conclusion

This paper summarized the results of the Fourth Workshop on Neural Generation and Translation, where we saw a number of research advances. Particularly, this year introduced a more rigorous efficiency task, and a new STAPLE task.

7 Acknowledgements


The efficiency shared task was partly funded from European Union's Horizon 2020 research and innovation programme under grant agreement No 825303 (Bergamot) and by the Connecting Europe Facility (CEF) - Telecommunications from the project No 2019-EU-IA-0045 (User-focused Marian). This work represents the authors' opinions, not necessarily those of the European Union.

We thank Amazon Web Services for its gift of credits to support the efficiency shared task evaluation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 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, Florence, Italy. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of EMNLP*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Belgium, Brussels. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.