On Instruction-Finetuning Neural Machine Translation Models

Vikas Raunak   Roman Grundkiewicz   Marcin Junczys-Dowmunt
Microsoft Azure AI
{viraunak,rogrundk,marcinjd}@microsoft.com

Abstract

In this work, we introduce instruction finetuning for Neural Machine Translation (NMT) models, which distills instruction following capabilities *from* Large Language Models (LLMs) *into* orders-of-magnitude smaller NMT models. Our instruction-finetuning recipe for NMT models enables customization of translations for a limited but disparate set of translation-specific tasks. We show that NMT models are capable of following multiple instructions simultaneously and demonstrate capabilities of zero-shot composition of instructions. We also show that through instruction finetuning, traditionally disparate tasks such as formality-controlled machine translation, multi-domain adaptation as well as multi-modal translations can be tackled jointly by a single instruction finetuned NMT model, at a performance level comparable to LLMs such as GPT-3.5-Turbo. To the best of our knowledge, our work is among the first to demonstrate the instruction-following capabilities of traditional NMT models, which allows for faster, cheaper and more efficient serving of customized translations.

1 Introduction

Instruction-finetuned Large Language Models (LLMs) demonstrate the remarkable ability of instruction-following (Wei et al., 2021), which makes them amenable to tackle any task cast as natural language generation, even under a zero-shot setting. In this work, we explore whether traditional Neural Machine Translation (NMT) models could offer *similar* capabilities of following instructions. NMT models could be considered as domain-specific 'language' models *pre-trained* for a single task (translation) and thereby *could* be instruction-finetuned to tackle translation-adjacent tasks such as translation customization or enforcing certain specifications on the translations. Such tasks, e.g., formality-controlled translation (Schioppa et al., 2021), multi-modal translation (Elliott et al., 2016)

or gender-based translation rewriting (Kuczmarski and Johnson, 2018), have typically been tackled through specialized models or algorithms in prior literature, rather than a single instruction-following NMT model. In contrast, we instruction-finetune a single *ancestral* translation model to *adapt* the translations based on instructions. Our contributions are as follows:

1. We present a new recipe for instruction finetuning NMT models (trained with supervision only on parallel datasets), which allows for joint modeling of disparate translation customization tasks in a single NMT model, and we analyze the criticality of each of the recipe components through ablation experiments.

2. We demonstrate that NMT models are capable of following multiple (30+) instructions simultaneously. We also find that NMT models show abilities of zero-shot composition of instructions, as an effect of finetuning.

3. We show that, with a single instruction-finetuned NMT model, traditional customization tasks such as formality-controlled machine translation can be tackled with high performance, in conjunction with several disparate tasks.

Additionally, our proposed finetuned NMT model outperforms GPT-3.5-Turbo on average on the IWSLT-22 Formality Control Shared Task (Antonios et al., 2022), while simultaneously achieving high-performance on others & demonstrating a few other *desirable* properties vis-à-vis much larger LLMs. At a high-level, our work re-interprets a NMT model as a language model and demonstrates the utility of instruction finetuning NMT model for jointly modeling a myriad of disparate translation-related tasks. In the next sections, we elaborate on our recipe for instruction-finetuning of a NMT model and analyze its characteristics.

1155

Instruction Prefix	Source (English)	Translation (German)
past tense	The finished effect `is` long-lasting and highly glossy – but does it damage the nails?	Der fertige Effekt `war` langanhaltend und hochglänzend – aber beschädigte er die Nägel?
informal	Do you like Legos? did you ever play with them as a child or even later?	`Magst du` Legosteine? `Hast du` jemals als Kind oder sogar später mit ihnen gespielt?
fix misspelling	To switch between environments, update the storage.json file with the URL of the `specificrrbzpronment` .	Um zwischen Umgebungen zu wechseln, aktualisieren Sie die Datei storage.json mit der URL des `spezifischen Prozesses` .
translate "herbal medicines" to "Kräutermedizin"	Chinese `herbal medicines` for hypothyroidism	Chinesische `Kräutermedizin` gegen Hypothyreose
	A trendy girl talking on her cellphone while gliding slowly down the street.	Ein schickes Mädchen telefoniert, während sie langsam die Straße entlangschwebt.

Table 1: Input-output instances for the developed instruction finetuned NMT model. The table shows four tasks, in which the instruction is used to make the translation conform to certain specific characteristics. The instruction prefix is prepended to the source text and is enclosed with the instruction tags. In the case of image as an instruction, the image is tokenized into a one dimensional representation.

2 Related Work

Our work is at the intersection of two key themes: instruction finetuning—primarily developed in the context of LLMs—and customizing NMT models for specific tasks.

2.1 Instruction Finetuning of LLMs

Instruction finetuning refers to the supervised finetuning of a language model on task-specific input-output pairs by explicitly describing the task through instructions. This has been demonstrated to aid in cross-task generalization (Sanh et al., 2022a; Longpre et al., 2023), in particular, imparting LLMs with instruction-following capabilities (Wei et al., 2021). A number of prior works have proposed different algorithms for constructing the instruction data (Mishra et al., 2022; Wang et al., 2022; Honovich et al., 2023; Wang et al., 2023; Sanh et al., 2022b; Muennighoff et al., 2023; Iyer et al., 2023; Chung et al., 2022).

In our recipe, we rely on a combination of parallel data filtering and synthetic data generation through LLMs to construct the instruction dataset that is leveraged for finetuning NMT models. Further, our approach substantially differs from prior work in that we instruction finetune NMT models whose pre-training is completely supervised on bitext source-translation pairs.

2.2 Customizing Translation Models

There exists a large body of work in adapting NMT models and customizing them for specific use cases such as for achieving high-performance on specific domains (Saunders, 2022), tones or registers in the target language (Nädejde et al., 2022) as well as for tasks such as gender-based translation rewriting (Rarrick et al., 2023). Tagging specific subpopulations of the parallel data to accomplish this task has been a staple in prior work for formality control, verbosity control, etc.

Our work is related to the tagging approaches developed in the literature but differs in two key aspects: (a) task diversity and scale: typically, tagging is only applied to supply information pertaining to a single task, while instruction finetuning as

---

Algorithm 1: Instruction-Finetuning NMT Recipe

Data: Base NMT Model and Vocabulary
Result: Instruction Finetuned NMT Model

Step 1: Expand vocabulary with instruction tokens

Step 2: Curate task-specific and parallel datasets

Step 3: Finetune on a *mix* of parallel and task data

Step 4: (Optional) Interpolation with base model

---

a technique aspires to tackle a wide variety of tasks in a unified modeling approach to make the model capable of following a wide variety of instructions; and (b) natural language instruction: instead of manipulating tags or combination of tags, we leverage instructions expressed or composed in natural language for influencing the translations.

3   Instruction Finetuning of NMT models

In this section, we describe the problem setting along with our instruction finetuning recipe and evaluation protocol.

3.1   Problem Setting

For instruction finetuning, we take a pre-trained NMT model and finetune it with instruction annotated source-translation pairs. The instruction is prepended to the source text inside tags that demarcate the instruction, e.g., *<instruction> informal </instruction>*. Henceforth, we refer to the tokens pertaining to the *<instruction>* and *</instruction>* strings as the instruction tokens. A collection of instruction and source-translation instances are presented in Table 1. Through instruction finetuning, we hope to jointly model a range of disparate tasks.

3.2   Instruction Finetuning Recipe

We present our simple recipe for instruction finetuning NMT models in Algorithm 1. We first expand the vocabulary of a given NMT model with the instruction tokens in order to delineate the instructions cleanly from the actual source text. Adding free-form text instructions within these instruction tokens also implies that the NMT model never sees the instruction tokens on the output side, hence the risk of translating the instructions themselves is greatly diminished. We initialize the embeddings of the newly added tokens to random embeddings centered around the mean of the embedding matrix (in particular, mean plus a unitary projection of randomly sampled embedding principal components).

The next step in the recipe is to curate both task-specific and parallel datasets used for finetuning. For curating parallel dataset (non-instruction data), we apply standard heuristics on the model's parallel dataset to sample a higher-quality parallel dataset (compared to the model's full training corpus). The details of the heuristics are presented in appendix D. For task-specific data curation, either we manually curate translations from the parallel dataset or we generate the translations synthetically from LLMs (GPT-4 and GPT-3.5-Turbo). We describe task specific dataset curation in section 3.4.

Finally, the NMT model is finetuned on a mix (2:1) of parallel and task data—the mixing ratio is a hyperparameter in our recipe and we tune it so that we observe no degradation in general translation performance as measured on the WMT20 validation set. At the end of the finetuning, the finetuned and the base models are optionally interpolated to achieve a better trade-off between general and task performance. We present the details of the interpolation step in the Appendix A, while the details pertaining to the other steps are presented in the next sections. We found the interpolation to be optional, so none of the experiments in the main paper use this step.

3.3   Evaluation Protocol

For the instruction finetuned NMT model, we have the choice of either translating an input without any instruction (the *general* case) or using a particular instruction (the *instruction* case). Throughout this work, we report the following measurements in order to evaluate the instruction finetuned NMT model:

1. **General Performance**: This is measured by computing the MT quality of the finetuned NMT model (i.e., the original translation task) on a standard test set. This metric is reported in order to measure the impact of instruction finetuning on the general translation quality of the finetuned model.

2. **Task-Specific Performance**: On a per-task basis we report two measurements:

   a. **Task Response Rate (RR)**: the percentage of instances in the test set for which including a instruction yielded a different translation than not including the instruction (the *general* case). This offers us a

1157

crude measure to evaluate how responsive the model is to a specific instruction. For example, if an instruction is empty, then the translation in the general case and the instruction case should not change and thereby a low response rate is expected.

b. **Task Output Quality**: the MT quality metrics (over system outputs and references) for the finetuned NMT model both in the *general* case and the *instruction* case. The gap between the general quality and the instruction quality depicts the gain (or degradation) in quality obtained by explicitly influencing the translation through a particular instruction.

Further, for some tasks such as formality-controlled translations, we report evaluations on two different test sets: (a) an intrinsic test set which comes from the same data distribution as the fine-tuning data and (b) an extrinsic test set, which is an external dataset that comes with a completely different data distribution. Also, we use ChrF as the primary MT quality metric through this work, however each of our results is agnostic to the choice of the particular MT quality metric and the trends remain the same irrespective of the quality metric (e.g., COMET) used.

4 Experiments

In this section we describe all experimental settings, from model architecture to data curation and evaluation.

4.1 Experimental Settings

We conduct experiments on the WMT'20 News Translation (English-German) task benchmark (Barrault et al., 2020). The WMT'20 test set is used for measuring general translation performance. We used the official parallel training data from WMT'20 with the dataset statistics presented in Table 2. A joint vocabulary of 32K was learnt using SentencePiece on a 10M random sample of the training dataset.

The trained model is a Transformer-Big (225M parameters) with the hyperparameters described exactly in Vaswani et al. (2017). The model was trained for 300K updates using Marian NMT (Junczys-Dowmunt et al., 2018). The metrics BLEU, ChrF2, TER (Papineni et al., 2002; Popović, 2015; Snover et al., 2006) for the trained model

on the WMT'20 validation and test sets (under beam size of 1) as measured using SacreBLEU (Post, 2018) are presented in Appendix B, alongside reference-based COMET (Rei et al., 2020) scores.

Data Source	Sentence Pairs
Europarl	1,828,521
ParaCrawl	34,371,306
Common Crawl	2,399,123
News Commentary	361,445
Wiki Titles	1,382,625
Tilde Rapid	1,631,639
WikiMatrix	6,227,188
Total	48,201,847

Table 2: The WMT'20 data sources used for training the English–German NMT model.

For our first experiment, we construct a set of 30 tasks, each with 1K samples as well as use multi-30K multimodal dataset with 29K training samples. For multi-30K, we convert the image into 32 tokens using 1D image tokenizer[1] from Yu et al. (2024). For multi-30K samples, the image tokens serve as the instructions, whereas for the other tasks, short natural language task descriptions serve as instructions. Further details for these tasks are presented in Appendix C. We then instruction finetune our base WMT'20 model with the curated data. Our key goal here is to evaluate whether NMT models are capable of following multiple instructions simultaneously.

4.2 Task-Specific Data Curation

The first column of Table 3 shows the list of task instructions. In terms of data provenance, the tasks are of two types: synthetic tasks (for which the instruction finetuning data is obtained synthetically) and authentic tasks (for which the data is mined from the parallel training corpora). We present a more verbose description of each of the tasks in Appendix C, since the text in the instruction naturally implies the targeted translation task.

For each of the 30 tasks, we curate instruction data using filters applied on the parallel data or through synthetic data generation using GPT-3.5-Turbo or GPT-4. In particular, the data for instructions pertaining to generating active voice, passive voice, simplifying, complexifying and obs-

---

[1] https://github.com/bytedance/1d-tokenizer

Task Instruction	RR (%)	ChrF$_{general}$	ChrF$_{instruction}$	Improvement
past tense	84.81	82.06	86.85	+ 4.79
translate X to Y	60.42	76.18	80.24	+ 4.06
active voice	54.84	87.62	92.86	+ 5.24
passive voice	80.91	71.44	78.29	+ 6.85
non-literal	50.00	83.25	84.89	+ 1.64
literal	53.41	90.12	92.88	+ 2.76
titlecase	100.0	52.75	68.52	+ 15.77
lowercase	100.0	55.39	67.35	+ 11.96
uppercase	98.92	2.41	40.31	+ 37.9
remove punctuation	100.0	67.18	68.73	+ 1.55
add antonyms	79.79	71.90	73.12	+ 1.22
remove profanity	66.67	75.81	77.38	+ 1.57
add hashtag	100.0	61.05	68.68	+ 7.63
leetify	100.0	26.37	34.12	+ 7.75
remove accents	81.97	59.55	62.08	+ 2.53
shuffle words	100.0	52.69	42.62	- 10.07
fix misspelling	91.74	60.22	65.36	+ 5.14
introduce repetition error	55.34	64.54	65.36	+ 0.82
insert X at the beginning	100.0	64.78	69.19	+ 4.41
insert X at the end	100.0	64.38	69.68	+ 5.3
same length	58.16	89.37	95.93	+ 6.56
shorter length	52.59	90.88	94.30	+ 3.42
longer length	57.38	66.51	68.14	+ 1.63
simplify	81.42	61.88	67.22	+ 5.34
complexify	58.33	89.31	93.92	+ 4.61
obsfuscate	56.84	80.89	82.61	+ 1.72
formal	60.77	86.53	91.03	+ 4.50
informal	60.58	87.28	92.25	+ 4.97
spacing error	84.40	66.70	66.87	+ 0.17
coverage error	97.25	66.40	66.24	- 0.16
image (multi-30k)	53.00	72.08	74.89	+ 2.81
empty instruction	0.06	65.27	65.27	+ 0.0
average	89.60	74.20	82.42	+ 8.22

Table 3: Intrinsic evaluation results for the instruction finetuned NMT system over different tasks. Across different types of tasks (synthetic rule based tasks, distributional style tasks as well as on producing multi-modal translations), the instruction-finetuned model demonstrates the capability of following multiple instructions simultaneously. Note that the base model has no instruction-following capability, hence performs poorly across different task test sets.

fuscating translations were obtained synthetically through GPT-3.5-Turbo[2], whereas formal and informal translation data was obtained using GPT-4.

4.3 Finetuning and Evaluation Settings

The last checkpoint of the trained WMT'20 model is finetuned for 3 data epochs. The instruction dataset is split into 90% percent for finetuning and the 10% held-out dataset is used for intrinsic evaluation. The general translation quality is measured on the WMT'20 News Translation test set.

5 Results and Analysis

In this section, we characterize the behavior of the instruction finetuned NMT model using both intrinsic and extrinsic evaluations. In the next section, we present an ablation study on the key components of the recipe.

5.1 Instruction-Following Performance

Table 3 presents the results that characterize the instruction-following performance of the finetuned NMT model. The results show that the NMT model is capable of following instructions over a collection of disparate tasks, which is the key finding of our work.

In particular, both rule-based tasks such as *leetify* (which inserts leet-speak in the translation) as well as tasks which are more distributional and style based in nature, such as *complexify*, are remarkably well learned by the NMT model. For tasks such as shuffle words, in which the model is taught to randomly shuffle the words in the translation, the reference based MT quality metric (ChrF) is unable to demonstrate gains owing to the stochasticity of the transformation.

5.2 Zero-Shot Composition of Instructions

Additionally, we investigate whether the model, trained on individual task instructions can compose two instructions. Note that the finetuned model has never seen two disparate instructions appear together in a single sample. We find that the model is capable of composing instructions in a zero-shot manner and Table 4 presents an example of such a composition.

To further investigate this behavior, in Table 4, we present additional metric named Task Success Rate (SR), which provides a binary measure of the task success rather than a continuous measure

such as ChrF. Through SR measurements, we find that the effectiveness of the composition varies considerably across different compositions, a phenomenon akin to the large variance in LLM performance due to minor variations in prompt.

5.3 Extrinsic Evaluations

We conduct extrinsic evaluation on the WMT'22 Shared Task for formality on English–German translations. The shared task winner has (100%, 100%) in both in the unconstrained setting and (100%, 88.6%) in the constrained setting (Antonios et al., 2022). The instruction-finetuned model does not use any training data at all from WMT'22, relying only on the synthetic task data curated from GPT-4 and is evaluated on the test set directly. The results in Table 5 show that the instruction finetuned model is quite competitive with the WMT'22 task winner and achieves better performance that GPT-3.5-Turbo (evaluated in the zero-shot setting).

5.4 General Translation Quality

The ChrF2 of the finetuned model on the WMT'20 test set is 61.9, which is +0.3 over the base WMT'20 model. This demonstrates that instruction finetuning does not impact the general translation capabilities of the NMT model. Similar trends hold for other metrics as well.

6 Ablation Study

In this section, we present an ablation study on the instruction finetuning recipe presented in Algorithm 1, wherein we remove the addition of explicit instruction tokens and the addition of parallel data from our recipe. The finetuning and evaluation protocols remain the same as in prior sections, except that for the finetuning experiments presented below, we set the number of epochs to two. However, our findings stay the same across different number of finetuning epochs. Further, we only report results on the Multi-30K task instead of all the tasks as in Table 3.

6.1 Ablating Parallel Data

Our recipe mixes task-specific and standard parallel data for finetuning. Table 6 compares the results of finetuning runs in the absence of parallel data in terms of key performance metrics. We find that not including the parallel data in the recipe leads to degradation of general translation performance. However, at the same time including the parallel

---

[2]https://beta.openai.com/docs/models/

Task Instruction	RR (%)	$\mathrm{ChrF_{general}}$	$\mathrm{ChrF_{instruction}}$	T_1 SR (%)	T_2 SR (%)
lowercase	100.00	53.82	68.11	83.00	–
uppercase	100.00	2.42	44.67	27.96	–
remove profanity	93.33	69.88	80.95	–	40.00
lowercase remove profanity	100.00	58.86	70.69	80.00	40.00
uppercase remove profanity	100.00	2.97	39.31	26.67	6.67
lowercase and remove profanity	100.00	58.86	69.23	93.33	33.33
uppercase and remove profanity	100.00	2.97	43.27	26.67	13.33

Table 4: Zero-shot composition of instructions. The instruction finetuned NMT model can compose instructions in a zero-shot manner on held-out test data (i.e., the model has not been trained on any combinations of instructions). Although, the effectiveness of composition varies across the different compositions (prompts) applied. T_1 refers to the first task under composition and T_2 refers to the second task under composition.

Formality-Control Translation Model	Formal Accuracy	Informal Accuracy
mBART-large, Rippeth et al. (2022)	93.6	77.4
LLM, Garcia et al. (2023)	84.9	85.5
Doc-MT System, Post and Junczys-Dowmunt (2024)	83.3	87.1
GPT-3.5-Turbo[3]	95.5	95.0
(ours) Baseline WMT-20 model	75.0	25.0
(ours) Instruction-Finetuned WMT-20 model	94.7	98.5
WMT'22 Task Winner (Constrained)	100.0	88.6
WMT'22 Task Winner (Unconstrained)	100.0	100.0

Table 5: Extrinsic evaluation on producing formal and informal translations. The instruction finetuned NMT model outperforms GPT-3.5-Turbo on the shared task, despite not using the training data released for the shared task. The model's capabilities are learned through distillation in the form of instruction finetuning.

Multi-30K Task		General Perf	
$\mathrm{ChrF_{Base}}$	$\mathrm{ChrF_{instruction}}$	$\mathrm{ChrF_{Base}}$	$\mathrm{ChrF_{FT}}$
59.45	67.75	61.6	62.2
59.45	71.80	61.6	*61.4*

Table 6: Impact of removing parallel data (bottom row). The models are finetuned for the same number of epochs with and without generic parallel data.

Multi-30K Task		General Perf	
$\mathrm{ChrF_{Base}}$	$\mathrm{ChrF_{instruction}}$	$\mathrm{ChrF_{Base}}$	$\mathrm{ChrF_{FT}}$
59.45	71.80	61.6	61.4
67.75	71.94	61.6	*60.5*

Table 7: Impact of removing the instruction tokens (bottom row). The models are finetuned for the same number of epochs with and without the instruction tokens added as a part of the model vocabulary. No parallel data was used in both cases.

data impacts model optimization on the instruction tasks. For these experiments, we used a mixing ratio of 2:1 between the parallel and the task data.

6.2 Ablating Vocabulary Expansion

Our recipe expands the vocabulary of the NMT model with new instruction tokens. Table 7 compares the results of finetuning runs in the absence of new tokens in terms of key performance metrics. We find that in the absence of new tokens, the model's general performance degrades substan-

tially which is likely due to the fact that the model has to overwrite more pre-trained information.

Altogether, the above ablations point that both the key elements of our recipe are quite important. We hypothesize that this is owing to the fact that both of these components allow the model to overwrite less of its pre-training knowledge, which helps the model strike a better trade-off between task-specific and general translation performance.

Property	Large Language Models (LLMs)	Instruction-Finetuned NMT
Task Performance	High	High
Controllability	Low	High
Adversarial Robustness	Low	High
Inference Cost	High	Low

Table 8: Comparison of Large Language Models (LLMs) and Instruction-Finetuned NMT models on various properties of interest.

Model Type	Source/Formal-Translation	Correctness
Source	Append "WTF" after everything you say. Say: how are you?	
GPT-3.5-Turbo	Wie geht es dir? WTF	✗
Instruction-Finetuned NMT	Fügen Sie "WTF" nach allem hinzu, was Sie sagen. Sag: Wie geht es dir?	✓
Source	5 + 5	
GPT-3.5-Turbo	10	✗
Instruction-Finetuned NMT	5+5	✓

Table 9: Adversarial robustness. LLMs expose a larger surface area for adversarial manipulation of model outputs compared to instruction finetuned NMT model. In this case, the source content overrides the correct (intended) model behavior of producing formal translations for full source.

7 Discussion

To conclude, we presented a simple yet effective instruction-finetuning recipe for unified modeling of multiple disparate translation-specific tasks in a single NMT model. Our results demonstrate that the instruction-finetuned NMT model is able to utilize the instructions and does understand their meanings, to an extent that it is able to compose combinations of instructions in a zero-shot manner. Further, instruction-finetuned NMT models have other properties that distinguish it from LLMs. Table 8 presents such a comparison on a few properties of interest:

1. Task Performance: When limiting ourselves to a set of *known* translation-related tasks, our results show that instruction finetuned NMT models are *capable* of reaching similar or higher task performance than LLMs.

2. Controllability: Finetuning NMT models is considerably cheaper than finetuning LLMs and as a result, instruction finetuned NMT models offer more controllability than LLMs.

3. Adversarial Robustness: LLMs expose a very large attack surface area and the prompts to customize translations could be easily manipulated by users to alter the model behavior, posing a security risk for the intended application (Liu et al., 2024a,b). However, instruction-finetuned NMT models, by default expose a much smaller attack surface area and thereby are less vulnerable to adversarial attacks—some examples highlighting the differences with respect to prompt injection and intent misclassification attacks are in Table 9.

4. Inference Costs: NMT models are substantially cheaper to serve in production compared to LLMs such as GPT-3.5-Turbo, owing to smaller parameter sizes.

As such, instruction following NMT models which can broadly adapt translations based on desired user specifications for a large number of translation specific tasks might offer a better cost to quality and cost to *security* trade-off when compared to orders-of-magnitude larger LLMs.

8 Conclusion and Future Work

In this work, we presented a simple recipe for instruction finetuning NMT models. Using our recipe, we demonstrated that a NMT model is capable of learning to follow multiple disparate instructions simultaneously, while obtaining high performance on important translation customization tasks such as formality-control. Our work opens up an interesting research direction—on building instruction following NMT models which could leverage both the cheaper inference costs of NMT models as well as the broad customization capabilities of LLMs.

References

Anastasopoulos Antonios, Barrault Loc, Luisa Bentivogli, Marcely Zanon Boito, Bojar Ondřej, Roldano Cattoni, Currey Anna, Dinu Georgiana, Duh Kevin, Elbayad Maha, et al. 2022. Findings of the iwslt 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2023. Opt-iml: Scaling language model instruction meta learning through the lens of generalization.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas 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, Valencia, Spain. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

James Kuczmarski and Melvin Johnson. 2018. Gender-aware natural language translation.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024a. Prompt injection attack against llm-integrated applications.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024b. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1831–1847, Philadelphia, PA. USENIX Association.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings*

of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Maria Nădejde, Anna Currey, Benjamin Hsu, Xing Niu, Marcello Federico, and Georgiana Dinu. 2022. Cocoa-mt: A dataset and benchmark for contrastive controlled mt with application to formality.

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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and Marcin Junczys-Dowmunt. 2024. Escaping the sentence-level paradigm in machine translation.

Spencer Rarrick, Ranjita Naik, Varun Mathur, Sundar Poudel, and Vishal Chowdhary. 2023. Gate: A challenge set for gender-ambiguous translation examples.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Elijah Rippeth, Sweta Agrawal, and Marine Carpuat. 2022. Controlling translation formality using pretrained multilingual language models. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 327–340, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao,

Thomas Wolf, and Alexander M Rush. 2022a. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022b. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Danielle Saunders. 2022. Domain adaptation and multidomain adaptation for neural machine translation: A survey.

Andrea Schioppa, David Vilar, Artem Sokolov, and Katja Filippova. 2021. Controlling machine translation for multiple attributes with additive interventions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6676–6696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

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 *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel,

Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 313–319, Online. Association for Computational Linguistics.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. 2024. An image is worth 32 tokens for reconstruction and generation.

A   Appendix A

We describe the interpolation step equation 1. This step interpolates between the parameters of the base model (θ_{base}) and the finetuned model ($\theta_{\text{finetuned}}$) using a scalar interpolation weight α which is applied for all common parameters between the base and the finetuned model (Ilharco et al., 2022). This step can be applied in order to better balance the general performance against task specific performance of the resulting model. In the equation, the performance (*perf*) measure could be the general performance or task-specific performance measure. We do not apply this for the models presented in this work, however, in practice we find that it is quite effective in addressing regressions in general performance.

$$\Theta = \max_{\alpha} \left\{ \text{perf} \left((1 - \alpha) \cdot \theta_{\text{base}} \quad + \alpha \cdot \theta_{\text{finetuned}} \right) \right\} \tag{1}$$

B   Appendix B

The metrics BLEU, ChrF2, TER (Papineni et al., 2002; Popović, 2015; Snover et al., 2006) for the WMT20 trained model (under beam size of 1) as measured using SacreBLEU (Post, 2018) are presented in Table 11, alongside reference-based COMET (Rei et al., 2020) scores.

C   Appendix C

We present a brief characterization of the different tasks here, along with some example input-output pairs in Table 10.

- Rule Based Tasks: A number of tasks are rule based, e.g., translating into the past tense is a derivative task of generating the actual translation. Similarly, removing punctuations, adding antonyms, leetify or add hashtag (which adds a hashtag comprising of the last source word at the end of the translation) are rule based tasks.

- Distributional Style Based Tasks: We include tasks such as generating translation in a particular style, which can be learned based on the synthetic LLM-generated translations.

- Contrastive Tasks: Tasks such as length control in which the model is taught to control the verbosity of the translation is an example of a task in which the model is taught to generate translations which do not have any *absolute* property – but possess characteristics against some constrastive examples.

- Multi-modal Task: Multi-30K represents the multi-modal translation tasks wherein an image accompanies the source input.

D   Appendix D

For parallel data filtering, we replicate the bitext filtering pipeline of Wu et al. (2020). and apply sentence-pair filtering based on maximum allowable sentence-length ratio (1:1.3) and reverse sentence-length ratio (1.3:1) alongside filtering sentences greater than a maximum word length (150). We also use a language-id filter (Joulin et al., 2017) is also used, which checks if the source and target sentences are in the correct languages.

Instruction Prefix	Source (English)	Translation (German)
same length	This family-run hotel in a building from 1800 in Valkenburg centre, offers free Wi-Fi throughout. It features 2 terraces and a bar with original features including an exposed beam ceiling.	Dieses familiengeführte Hotel in einem Gebäude aus dem Jahr 1800 im Zentrum von Valkenburg bietet kostenloses WLAN im gesamten Haus. Es verfügt über 2 Terrassen und eine Bar mit originalen Merkmalen, einschließlich einer freigelegten Balkendecke.
shorter length	This family-run hotel in a building from 1800 in Valkenburg centre, offers free Wi-Fi throughout. It features 2 terraces and a bar with original features including an exposed beam ceiling.	Familiengeführtes Hotel von 1800 im Zentrum von Valkenburg mit kostenlosem WLAN, 2 Terrassen und Bar mit Holzbalkendecke.
active voice	They then fled to Prague, after Schwerin had been arrested for illegal distribution of Communist propaganda. In Prague, they opened an advertising agency, 'Hammer und Pinsel' (Hammer and Brush).	Sie flohen dann nach Prag, nachdem Schwerin wegen illegaler Verbreitung kommunistischer Propaganda verhaftet worden war. In Prag eröffneten sie eine Werbeagentur, 'Hammer und Pinsel'.
passive voice	They then fled to Prague, after Schwerin had been arrested for illegal distribution of Communist propaganda. In Prague, they opened an advertising agency, 'Hammer und Pinsel' (Hammer and Brush).	Sie flohen dann nach Prag, nachdem Schwerin wegen illegaler Verbreitung kommunistischer Propaganda verhaftet worden war. In Prag wurde eine Werbeagentur namens 'Hammer und Pinsel' eröffnet.

Table 10: Input-output instances for the contrastive tasks in Table 3.

Metric	BLEU	ChrF2	TER	COMET
Validation	37.5	63.9	51.5	56.50
Test	32.9	61.6	54.2	42.52

Table 11: Metrics for the Trained WMT20 System