Controlling Machine Translation for Multiple Attributes with Additive Interventions

Andrea Schioppa* and **Artem Sokolov** and **David Vilar** and **Katja Filippova**
Google Research
{arischioppa, artemsok, vilar, katjaf}@google.com

Abstract

Fine-grained control of machine translation (MT) outputs along multiple attributes is critical for many modern MT applications and is a requirement for gaining users' trust. A standard approach for exerting control in MT is to prepend the input with a special tag to signal the desired output attribute. Despite its simplicity, attribute tagging has several drawbacks: continuous values must be binned into discrete categories, which is unnatural for certain applications; interference between multiple tags is poorly understood. We address these problems by introducing vector-valued interventions which allow for fine-grained control over multiple attributes simultaneously via a weighted linear combination of the corresponding vectors. For some attributes, our approach even allows for fine-tuning a model trained without annotations to support such interventions. In experiments with three attributes (length, politeness and monotonicity) and two language pairs (English to German and Japanese) our models achieve better control over a wider range of tasks compared to tagging, and translation quality does not degrade when no control is requested. Finally, we demonstrate how to enable control in an already trained model after a relatively cheap fine-tuning stage.

1 Introduction

Some modern machine translation (MT) applications require fine-grained control along multiple attributes, and such mechanisms also increase the users' trust in scenarios when the system speaks on their behalf (Prabhumoye et al., 2021). For example, MT applications like video subtitling in streaming, video conferencing, online education and speech MT require that one can control the length and monotonicity of the translation, setting clear constraints on the output. In open-domain MT, it is unlikely that such constraints are known or can be inferred from the source to generate an appropriate translation. However, the uncertainty around the desired register, style or politeness level of the translation could be resolved by providing users with an explicit option to control such attributes. This in turns increases the MT system's trustworthiness by providing an explicit contract (Jacovi et al., 2021), formulated as "whenever there is an ambiguity, we enable users' agency".

A standard method to exert control over MT outputs is the *tagging* approach, where an explicit token is prepended to the source sentence or output hypothesis to signal the desired attribute of the output (Kobus et al., 2017; Sennrich et al., 2016; Johnson et al., 2017). While such tags do enable certain level of control, discrete tags, by their nature, allow only for coarse-grained control and require that attributes with continuous values, like monotonicity or length ratio, are binned. For example, Lakew et al. (2019) used only three tags to control translation length, which would arguably be too coarse for many practical applications. Also, chaining multiple tags may become cumbersome and, more importantly, the interference between tags and the effect of their ordering have not yet been extensively studied.

An additional desideratum for systems enabling attribute control is how efficiently they can be realized. For deployed MT engines, (re-)training a model for every attribute is unrealistic, due to the associated costs in time and computational power. Therefore, having a light-weight intervention, materialized as a small number of tunable parameters, would considerably improve the practicality of attribute-enabled systems.

In this paper we introduce additive vector-valued interventions which allow for fine-grained, combinable and fine-tunable control of translations, addressing all of the points above. We propose two implementations of vector-valued control: 1) one

* Google AI Resident.

attribute embedding vector with the control direction and strength regulated by a multiplicative scalar factor, appropriate for continuous attributes, and 2) separate embedding vectors for each discrete attribute value, each with tunable multiplicative strengths. The attributes' embeddings are additively combined with the encoder's last layer representation and are used by a subset of the decoder layers through the source-attention mechanism.

Compared with the tagging approach, the control intervention residing in vector spaces has three advantages: 1) It avoids the coarse binning inherent to tagging and enables a more fine-grained, wider-range and precise control of translations, especially around bin boundaries. 2) It simplifies simultaneous control for multiple attributes via a linear combination of control interventions for each attribute, with control strength defined by multiplicative scaling factors. 3) For some attributes it allows for enhancing neural MT models trained without controllability via fine-tuning of intervention vectors.

Our contributions are as follows:

1. We propose a novel mechanism to control different translation attributes and evaluate it on three important use cases: length, politeness and monotonicity for translation into German and Japanese (from English).

2. In all the three use cases, the ability to control attributes comes at no cost in translation quality. In fact, including explicit politeness information, the evaluation scores improved as compared to strong baselines (+0.6 BLEU points for German and +2.5 for Japanese).

3. Given a system trained on data without attribute annotation, we demonstrate that we can add a control component to it, needing only 20% of the original training time. The level of control is not on-par with a full training pass, but the performance is still similar to the tagging approach.

2 Related work

The tagging approach for controlling translations has been used for multiple purposes: to indicate the target language in multilingual NMT (Johnson et al., 2017); to produce translations in more natural language by tagging data provenance, back-translated or natural (Caswell et al., 2019); to control gender (Kuczmarski and Johnson, 2018; Elaraby et al., 2018); to indicate source domains in multi-domain NMT (Kobus et al., 2017). Closer to our applications, tags can control length (Lakew et al., 2019), and formality of translations from English to German (Sennrich et al., 2016) and to Japanese (Yamagishi et al., 2016; Feely et al., 2019), as well as from French to English (Niu et al., 2018). Stergiadis et al. (2021) experimented with a pair of tags to control domain and provenance.

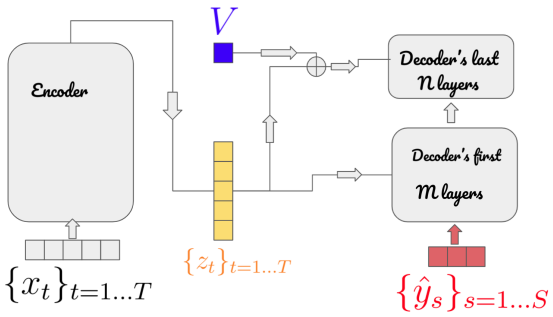


Figure 1: Model diagram.

Closely related to controllable generation is work on monolingual style transfer: Krishna et al. (2020), Riley et al. (2021) and Niu et al. (2018). In contrast to these papers, we use a classifier of the target side for labelling the controlling attribute.

3 Additive control

3.1 Base Transformer model

The Transformer model (Vaswani et al., 2017) consists of a decoder $\mathcal{D}$ and an encoder $\mathcal{E}$; the latter takes the input tokens $\{x_t\}_{t=1...T}$ and produces an intermediate encoded representation $z = \{z_t\}_{t=1...T} \in \mathbb{R}^d$. Layers of $\mathcal{D}$ then decode this representation z into a target sentence $\hat{y} = \{\hat{y}_s\}_{s=1...S}$. The decoding process is carried out in an auto-regressive way: at each time step t the decoder uses the previously generated output tokens $\{\hat{y}_s\}_{s \leq t}$ and accesses z through the attention mechanism.

3.2 Control-induced Transformer model

We propose to achieve control in the encoder's intermediate space by intervening with a perturbation of the representations z. For each attribute a to control we define an intervention vector V_a of the same dimensionality as z_t, which is added to all outputs z_t of the encoder $\mathcal{E}$. Defining $V = \sum_a w_a V_a$, the new hidden representation at each step becomes $\tilde{z}_t = z_t + V$. Note that w_a is a continuous weight that can be used as a "dial" to tune the strength

of the intervention for each attribute. The additive approach is motivated by the following desiderata:

1. It is an approach with a clear algebraic structure that covers multiple attributes in an interpretable manner.

2. It ensures the existence of a neutral state and the possibility of only modifying a subset of attributes.

3. It is permutation invariant, i.e. we did not wish to have a dependence on the specification order of the attributes (cf. tagging approach).

At training time we need an annotation of the bilingual sentence pairs to train the representations V_a, but we do not require the full training data to be annotated. For unannotated training pairs we simply set the vector V to zero. This makes the approach specially attractive if the desired attribute is costly to annotate, be it because of the need of expensive human annotation, fuzzy definition of the attribute itself or an expensive classifier to run.

Similarly, at inference time if the user does not want to control attribute a, V_a can be set to $\mathbf{0}$, the "neutral" vector. Setting the full intervention $V = \mathbf{0}$, which we denote the "neutral mode", we recover the behaviour of the initial underlying model, guaranteeing a fall-back to baseline performance (assumed to be of acceptable quality).

We also experimented an architecture in which V was prefixed to the the embeddings corresponding to the input sequence $\{x_t\}_{t=1...T}$ instead of being added to the encoder representation $\{z_t\}_{t=1...T}$, which more closely resembles the tagging architecture. This approach however resulted in models with a degraded translation quality for the continuous attributes, and thus we focus the discussion on the additive approach.

3.3 More efficient realization

We additionally considered a modification of the approach described in the previous section where the shifts $z_t + V$ are only accessible by the last N layers of the decoder, see Figure 1. For example, the first decoder layers have access to the standard, non-modified, z_t encoder representation through the attention mechanism, while the last layers access the modified $\tilde{z}_t$. This modification allows for faster training and small footprint fine-tuning, as the weights of the first decoder layers are kept fixed. From a model interpretability point of view, we can use this modification to understand which layers process a specific syntactic / semantic attribute, as an attributes-informed version of layer probing used to analyze Transformer encoder-only models (Tenney et al., 2019).

3.4 Attribute representation

In this work we considered three different attributes for control, but the approach can naturally be generalized to other attributes.

Length (L) For length control the confounding factor is that longer inputs would generate longer translations. Thus, instead of aiming to control the output length directly, we control the ratio r between the output and input lengths, both computed after tokenization and subword splitting. For this attribute the weight w_l corresponds to the ratio r, and the system learns the length control embedding V_l.

Politeness (P) Although politeness is an inherently discrete attribute, we also introduce a continuous feature representation (P_d vs. P_c). The discrete feature uses a separate embedding for each politeness level i, i.e. we train a different V_{p_i} vector for each politeness level. For the continuous feature we fix the weights w_{p_i} of the different levels, and the system trains a single politeness embedding vector V_p.

Monotonicity ($\mathbf{M}_{0.1}$) We understand monotonicity as the closeness of the word order in the target sentence to the word order in the source sentence. We formally define monotonicity as the strength of the off-diagonal alignment deviations, inspired by the `fast_align` model (Dyer et al., 2013). For a translated pair $s = (s_{\text{input}}, s_{\text{target}})$ and an alignment $\{(i, j)\}$ between the token positions $i \in \{1, \cdots, n\}$ of the input sentence s_{input} and $j \in \{1, \cdots, m\}$ of the target sentence s_{target}, we define the deviation strength:

$$\delta(s) = \frac{1}{\#\{(i,j)\}} \sum_{\{(i,j)\}} \left| \frac{i}{n} - \frac{j}{m} \right|, \qquad (1)$$

where $\#\{(i, j)\}$ denotes the cardinality of the alignment. In the completely monotonic case, having $n = m$ and $\{(i, j)\}$ being a strictly increasing bijection, $\delta(s)$ would be zero; in the general case, the lower $\delta(s)$ is, the higher the monotonicity between the input and the translation. To annotate $\delta(s)$ in the training data we used `fast_align`, and this is fed into the system as the weight w_m.

However, if $\delta(s)$ is small the resulting representation could potentially "collide" with the neutral state $V_m = 0$; we therefore use the shifted representation $w_m = \delta(s) + k$; we found that a small shift like $k = 0.1$ works well to avoid a collision.

For all attributes, we looked for the minimum number of decoder layers that have access to the representation with interventions that would work for all three attributes and found that two layers was the smallest value (even for length, the simplest attribute, one layer was not enough).

4 Experiments

We evaluate our control approach on two language pairs (English-to-German and English-to-Japanese), and three different attributes (length, politeness and monotonicity of the generated translations). We verify the validity of the approach regarding the general translation quality and for each controlled attribute individually. We end this section with experiments on fine-tuning additive control models from pretrained baseline models that were trained without any attribute annotation. For reproducibility, we include setup details in §A.4.

4.1 Datasets and baselines

For EN ⇒ DE we trained on the WMT17 dataset, using newstest2016 as the development set and newstest2017 as the test set (Bojar et al., 2017). In order to test the behaviour on an out-of-domain setting, where the distribution of the controlled attribute may vary from the training data, we also evaluate our methods on a subset of OpenSubtitles. For EN ⇒ JA we trained and evaluated on JESC (Pryzant et al., 2018). All the reported results use SacreBLEU (Post, 2018)[1].

4.2 Model configuration and training

We reimplemented the standard Transformer architecture (Vaswani et al., 2017) in JAX (Bradbury et al., 2018), using the neural network library Flax (Heek et al., 2020). All our models correspond to the Base Transformer configuration (Vaswani et al., 2017).

For training our additive models we label the whole corpus with the corresponding attributes and use the standard cross-entropy loss. However, to encourage the additive model to learn to produce good translations in the Neutral mode, we randomly mask each attribute independently with a 20% chance. We also trained an improved tagging baseline Tag_{mask} where tags are masked at a 20% rate so that it approximates the Neutral mode of the additive model. As there was a 2.7% relative difference in BLEU[2] caused by the different order of tags we also trained a mode Tag_{inv} where, *additionally* to being randomly masked, tags are also shuffled to achieve permutation invariance. For binning the continuous attributes of the tagging models we used five buckets for length and three for monotonicity.

Model	Mode	BLEU
Base	-	27.11
$\text{Tag}(\text{L}, \text{P}_d, \text{M})$	Oracle	26.58
$\text{Tag}(\text{L}, \text{M}, \text{P}_d)$	Oracle	27.32
$\text{Tag}_{\text{mask}}(\text{L}, \text{M}, \text{P}_d)$	Neutral	26.92
$\text{Tag}_{\text{mask}}(\text{L}, \text{M}, \text{P}_d)$	Oracle	27.12
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Neutral	27.02
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Oracle	26.98
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Neutral	26.92
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	26.99
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Neutral	27.43
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	27.76

Table 1: BLEU scores on WMT EN-DE. The difference between the best and worst tagging models, where only the tag order is changed, is statistically significant ($pval < 10^{-10}$).

4.3 Translation quality results

The main goal of the additive interventions is to achieve precise control of the desired attributes. As such, translation quality as measured by standard metrics may degrade if we keep the references fixed (e.g. generating a translation with an informal politeness level when the reference is polite). To this end we also analyse the effect of control-enabled models on general quality to ensure their performance is on par with the baseline models. We contrast the Neutral and the Oracle modes where the latter corresponds to a realistic scenario where the user knows what attribute value the output should have. A good control model is expected to take advantage of the Oracle information and improve its performance.

When presenting the results, the additive models are denoted by Add with the enabled attribute fea-

[1] Configuration signatures in §A.3.

[2] Reported as (max-min)/mean of BLEU scores.

Model	Method	BLEU
Base	-	15.14
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Neutral	15.02
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Oracle	17.52
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_{c,})$	Neutral	14.64
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	18.04
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Neutral	14.92
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	17.60

Table 2: BLEU scores on JESC EN-JA.

tures indicated between brackets; the subscript in Add_2 means that the intervention was applied only on the last two decoder layers. Models using tags are denoted by Tag.

EN ⇒ DE (Table 1): The performance of the additive model in the Neutral mode is very close to or even better than the Baseline and the Neutral variants of the tagging models. Thus, training a control-enabled model does not hurt translation performance even in the case where the attribute values are left unspecified.

For tagging, by ordering the tags differently, we get results between 26.58 and 27.32 points, which indicates that the tag order may require additional fine-tuning. (L, M, P) produced the best result while the permutation trick for alleviating order effect (i.e. Tag_{inv}) helped but did not solve the problem completely. It is worth noting that using masking to support the Neutral mode works well both with continuous and tagging models.

EN ⇒ JA (Table 2): The performance of the best additive model in Neutral mode suffers a reduction of 0.2 BLEU in comparison to the baseline, similar to the $\text{Tag}_{\text{inv}}(\text{L,M,P}_d)$. Importantly, moving to the Oracle mode regains up to 2.9 BLEU over the baseline which is a better improvement than what the tagging model achieves in the same Oracle mode.

4.4 Controlling length

We turn to evaluating length control and show that the continuous approach yields a more fine-grained and robust control than tagging.

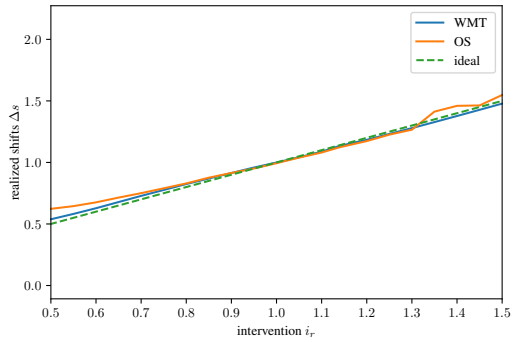
For this analysis we compute the ratio r of the source sentence length with respect to the reference length, and ask the model to produce a longer or shorter translation by a multiplicative intervention, i.e. replacing r with $r \times i_r$. For example $i_r = 1.0$ corresponds to asking the model to match the length of the references, while $i_r = 0.9$ to make translations 10% shorter than the references. We can then measure the effectiveness of length control by regressing the length of translations over the length of references to obtain a realized length shift Δs as a function of i_r, where an ideal control would achieve $\Delta s = i_r$.

We plot results for the model $\text{Add}(\text{L,M}_{0.1}\text{,P}_c)$ in Figure 2a. To measure the degree of distributional robustness we also measured the realized shifts on a test set from OpenSubtitles as an out-of-distribution test set. As the models were trained on WMT17, on OpenSubtitles ideally one should obtain the same length control we achieved on WMT17 with the same Δs resulting from the same i_r.

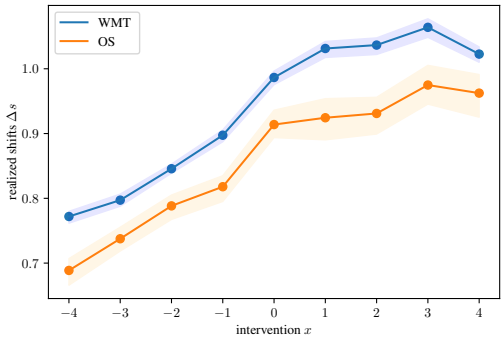
We illustrate how the BLEU score changes with the value of intervention in Figure 2c, where the interventions show a graceful degradation of BLEU (about 2.3 BLEU points to accommodate a 10% length change). To make sure that the additive control reformulates sentences in a sensible way and not simply repeats or trims tokens and is not limited to simple word-level modifications, we considered a naive baseline, *rewriter*, that takes the translations from the neutral mode and rewrites them to the resulting desired length either by truncating or by repeating tokens cyclically from the beginning till reaching the desired length. We compare the BLEU scores of this rewriter with our proposed model in Figure 2d: for German the difference is positive for i_r in the wide range $[0.75, 1.3]$ and for Japanese the range is even wider[3]. We provide exemplars of changing the length for German and Japanese in §A.13.

Comparison to tagging. For tagging we can shift length in incremental steps by shifting the tag bucket id (corresponding to the reference), e.g. $id + x$ for $x \in \{-4, \cdots, +4\}$ and clipping it to stay in the range of available buckets. Here $x = 0$ would correspond to the (length) Oracle mode. Note that tagging achieves a much smaller range of effective length control (Figure 2b) than our continuous method and that Δs is not a monotonically increasing function of x. For the out-of-distribution robustness we compared the realized shifts of tagging and the continuous method using the test sets of OpenSubtitles and WMT for Ger-

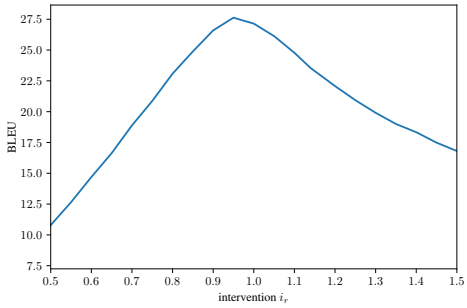
[3]Out of the plotted range the model and the rewriter become equivalent.



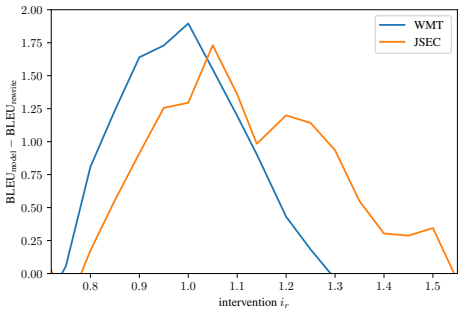
(a) Length control: realized shifts by Add model.



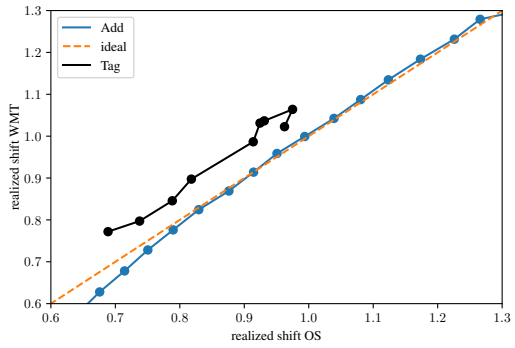
(b) Length control: realized shifts by Tag model with confidence bands due to bucketing.



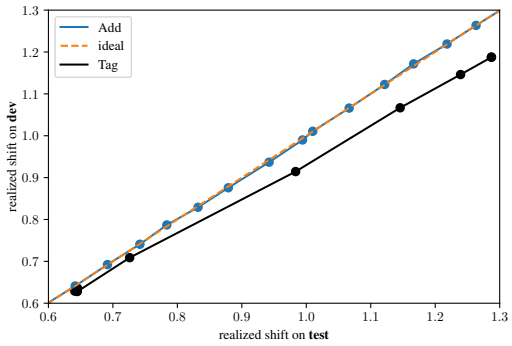
(c) Length control: BLEU on WMT as a function of i_r.



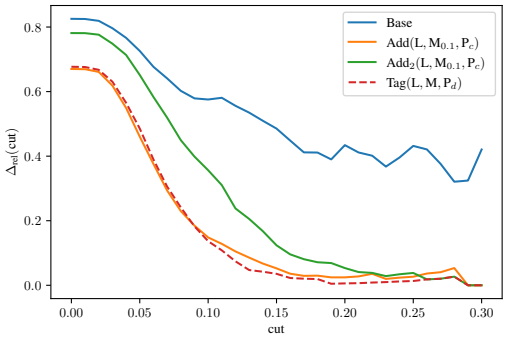
(d) Length control: BLEU comparison to the *rewriter* postprocessor for German (WMT) and Japanese (JESC).



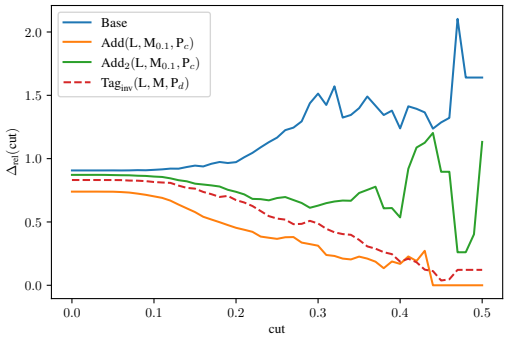
(e) Length control: out-of-distribution comparison for German.



(f) Length control: comparison on test/dev set for Japanese.



(g) Increasing monotonicity: WMT.



(h) Increasing monotonicity: JESC.

Figure 2: Control Evaluation.

man (Figure 2e) and the test and dev set of JESC for Japanese[4] (Figure 2f). We see that the continuous method gives consistent, close to ideal, shifts for the same interventions, while tagging is affected by the distribution shift.

4.5 Controlling politeness

We now focus on controlling translation politeness and formality for two languages that mark these registers: German with two formality levels and Japanese with a more developed hierarchy of speech registers.

EN ⇒ DE We annotated German politeness using the ParZu parser and the lexical rules from (Sennrich et al., 2016), which mostly look at the German 2nd person pronouns 'Sie' (polite 'you') and 'du' (informal 'you') and the corresponding verbs. Because WMT contains a very small amount of the *informal* class, for evaluation purposes we used the test set for OpenSubtitles[5]. We introduced a third annotation level, *unknown*, as a sink for the examples that the rule-based classifier assigns neither to *polite* nor to *informal*; during translation we found that introducing and enforcing the *unknown* mode results in a frequent switch to the indefinite German pronoun "man" that corresponds to impersonal speech. For example, the English sentence "What would you like to eat?", would be translated into the *unknown* politeness as "Was will man essen?" ("What would one like to eat?").

We found that politeness for the additive models can be controlled with similar results using either discrete P_d or continuous P_c. As P_c relies on a lower 1-dimensional latent representations, we focus on reporting results for the P_c representation. For the multipliers w_{p_i} we used values $\{0.5, 1.0, 1.5\}$ for *unknown*, *polite* and *informal* respectively. We did not aim to tune these multipliers (e.g. by treating them as hyper-parameters or model parameters), because our goal was to show that as long as there is some separation between the values the model can learn to generate different formalities, irrespective of a formality ranking order (e.g. having *unknown* in between *polite* and *informal*).

To evaluate the quality of politeness control, as in previous works (Sennrich et al. (2016), Feely et al. (2019)), we measure BLEU improvements on different splits of the OpenSubtitles test set (Table 3). Note that in all the additive models the Oracle mode leads to substantial improvements, especially on the informal split of the test data. Moreover, one can further improve the results by tuning a small length intervention (denoted by L-Fin) on top of the length oracle[6], which is probably effective because evaluation here happens out-of-distribution. In the supplementary materials (Table 9) we report the results of applying the politeness classifier on the generated translations. In the first exemplar in Table 5 we give an example of changing the politeness level in German to match the reference. For Japanese we include exemplars in the supplementary material in Table 15.

Model	**Mode**	all	*unknown*	*polite*	*informal*
Base	-	19.32	20.75	20.59	13.31
$\mathrm{Tag}(\mathrm{L}, \mathrm{M}, \mathrm{P}_d)$	Oracle	21.99	22.05	24.94	20.47
$\mathrm{Tag}(\mathrm{L}, \mathrm{P}_d, \mathrm{M})$	Oracle	21.17	21.14	23.66	20.21
$\mathrm{Tag}_{\mathrm{mask}}(\mathrm{L}, \mathrm{M}, \mathrm{P}_d)$	Neutral	19.66	21.05	21.01	13.78
$\mathrm{Tag}_{\mathrm{mask}}(\mathrm{L}, \mathrm{M}, \mathrm{P}_d)$	Oracle	21.50	21.67	23.29	20.09
$\mathrm{Tag}_{\mathrm{inv}}(\mathrm{L}, \mathrm{M}, \mathrm{P}_d)$	Neutral	19.41	20.77	21.21	13.47
$\mathrm{Tag}_{\mathrm{inv}}(\mathrm{L}, \mathrm{M}, \mathrm{P}_d)$	Oracle	21.38	21.42	23.35	20.33
$\mathrm{Add}(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	Neutral	19.99	21.82	22.41	12.33
$\mathrm{Add}(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	Oracle	21.55	21.95	24.13	18.93
$\mathrm{Add}(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	L-Fin	21.97	22.33	24.73	19.39
$\mathrm{Add}_2(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	Neutral	20.33	21.98	23.72	12.83
$\mathrm{Add}_2(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	Oracle	21.70	22.05	24.26	19.32
$\mathrm{Add}_2(\mathrm{L}, \mathrm{M}_{0.1}, \mathrm{P}_c)$	L-Fin	22.32	22.6	24.55	20.15

Table 3: BLEU scores of a WMT-trained model on OS by politeness split. Sizes: all: 4566, *unknown*: 3617, *polite*: 276, *informal*: 673.

For Japanese politeness and formality levels we re-implemented the rules of (Feely et al., 2019) introducing a fourth category *unknown* in addition to the original three classes *informal*, *formal* and *polite* (§A.9). To first approximation, the *polite* level is characterized by specific verb endings, e.g. です or ます, while the *formal* one is characterized by honorific expressions, e.g. ございます. The multipliers we used can be found in the supplementary materials (Table 10). We see that controlling politeness improves BLEU scores on every split when the rule-based feature is supplied (Table 4).

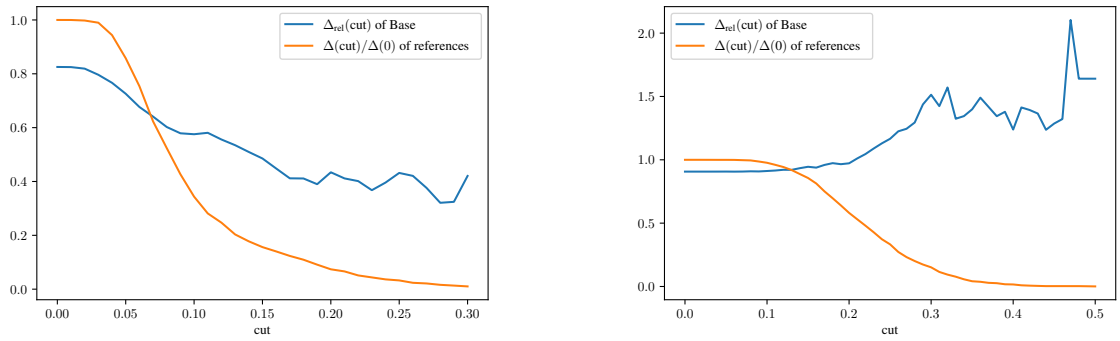
4.6 Controlling monotonicity

In this task we simulate a use case where we need the NMT system to produce translations of increasing monotonicity, having in mind applications like interpreting or lecture translation. Here the intervention consists in supplying to the model a desired value δ for the $\delta(s)$ of Equation 1.

[4] Some tags lead to the same shift, so we get < 9 points.

[5] The dataset was also considered by Sennrich et al. (2016).

[6] $i_r \in [0.9, 1.1]$ tuned on the dev set.



(a) WMT

(b) JESC

Figure 3: Comparison of the translation monotonicity to references for translation into German (WMT) and Japanese (JESC).

Model	**Method**	*unknown*	*informal*	*polite*	*formal*
Base	-	14.60	14.92	15.61	21.95
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Neutral	14.51	15.66	15.06	27.87
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	Oracle	14.42	19.63	20.03	52.85
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Neutral	15.16	14.25	17.24	38.31
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	15.60	19.32	20.28	53.28
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Neutral	16.11	15.42	15.73	20.44
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	Oracle	16.31	17.79	18.97	45.15

Table 4: BLEU scores on JESC by politeness split. Sizes: *unknown*: 1176, *informal*: 308, *polite*: 508, *formal*: 8.

input	You're good, aren't you?
reference	Dir geht's gut, nicht?
neutral	Sie sind gut, sind Sie nicht gut?
match reference politeness	Du bist gut, bist du nicht?
input	It was a first for the 58-year-old.
reference	Für den 58-Jährigen war es eine Premiere.
make more monotone	Es war ein erster für den 58-Jährigen.
match reference monotonicity	Für den 58-Jährigen war dies eine erste.

Table 5: Controlling politeness and monotonicity in German translations.

Non-monotonicity measure. We introduce as a measure of non-monotonicity for a set of translations pairs S

$$\Delta(S) = \sum_{s \in S} \text{len}(s_{\text{target}}) \times \delta(s),$$

which intuitively measures by how many positions the translation deviates from the input sentence[7]. To measure the fraction of translations surpassing the references in terms of token displacements we introduce the relative non-monotonicity

$$\Delta_{\text{rel}}(\text{cut}) = \frac{\Delta(\{s' = (s_{\text{in}}, s_{\text{out}}) : \delta(s') \geq \text{cut}\})}{\Delta(\{s = (s_{\text{in}}, s_{\text{ref}}) : \delta(s) \geq \text{cut}\})},$$

which allows us to take a "snapshot" at different thresholds for cut, comparing generated outputs with references.

To make this more clear we report the non-monotonicity measure for the Base model, starting with German. Looking at Figure 3a, it becomes clear that the Base translating to German produces translations that are more monotone than the references and that $\Delta_{\text{rel}}(\text{cut})$ decreases with cut. For comparison, we have also plotted the ratios $\Delta(\text{cut})/\Delta(0)$ for the references to highlight how cut affects the distribution of $\text{len}(s_{\text{target}}) \times \delta(s)$ for the references; intuitively for German lots of "mass" for the references is concentrated at low values of $\delta(s)$. We report the same for the Base model translating into Japanese in Figure 3b. Here the situation is different – while as before the Base produces translations that are more monotone than the references, the rate of drop is slower than for German and then the trend reverses at about $\text{cut} = 0.2$ where $\Delta(\text{cut})/\Delta(0) = 53\%$. Japanese references also put more mass on higher values of cut than the German ones; this should not be surprising, as English and German languages are SVOs while Japanese is SOV, so more re-ordering are necessary to translate into the latter.

Evaluation of control with respect to monotonicity. In Figure 2g we compare a few EN ⇒ DE control-enabled models on the task of monotonicity control of translations. All the models produce more monotone translations compared to the baseline and there is no significant difference between tagging and additive control. However the model Add_2 has a smaller effect than the model

[7] A detailed derivation is in A.10.

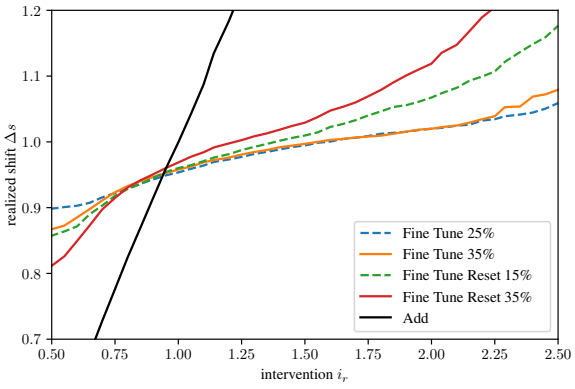


Figure 4: Fine-tuning length control.

Add in improving monotonicity, probably indicating that monotonicity is a harder attribute benefiting from the interplay between more layers. A similar conclusion holds for Japanese (Figure 2h). Here it might be interesting to note that Base, for large values of cut, produces translations that are less monotone than the references and that even the simpler Add_2 helps to reduce this effect. In the second exemplar in Table 5 we give an example of increasing the monotonicity compared to the reference and of matching the reference's alignment score. For Japanese we supply examples in §A.13. In terms of *decreasing* monotonicity we found that the continuous approach is more fine-grained; more details are given in §A.11.

4.7 Learning to control attributes with fine-tuning

Obtaining controllable models with fine-tuning a baseline model is important to reduce costs of developing attribute-specific models and reduce memory, ideally allowing to override a (small) subset of parameters of the main model already in memory.

We focused on the direction EN $\Rightarrow$ DE starting from the checkpoint of the Base model and we were able to learn politeness and length control, while monotonicity proved to be a harder attribute to bootstrap from the baseline model. Simultaneously we aimed at learning joint attribute control with a minimal number of parameters – learning just the attribute embedding(s) and either fine-tuning the last two layers of the decoder or resetting them to a random initialization, both affecting about 13.9% of the original model parameters.

In Figure 4 we report results at two time points during the training: the first was chosen when we saw an early indication of achieving control and the second when the control results had stabilized. Here Δs is an increasing function of i_r even though not close to the ideal $\Delta s = i_r$ as for the Add model; for the model with (without) resetting at about 15% (20%) of the original training time one can increase length by about 17% (5%), and one can decrease length by about 15% (10%). Regarding politeness, for the first time point the gains on OpenSubtitles between the Neutral and the Oracle mode are already relatively close to those obtained with training from scratch (§A.12). Overall, BLEU scores remain close to those of the model trained from scratch (e.g. on WMT 26.78 in Neutral mode for the model without resetting).

5 Conclusions

We propose a novel approach for controlling NMT system with respect to multiple attributes. This approach has several advantages: first, it uses *interpretable* additive interventions, where each attribute has a "control" subspace in latent space; second, it allows to control any subset of attributes while still generating good quality translations in the absence of any attribute intervention; third, it results in a more fine-grained and robust control of continuous attributes compared to the common tagging approach without the necessity of committing to a choice of buckets for continuous features; finally, it allows for a more efficient fine-tuning procedure where attribute control can be introduced by affecting a smaller subset of the original model parameters. We show-cased the flexibility of the approach by controlling length, politeness and monotonicity of generated translations from English into German and Japanese. Future directions of work include: 1) learning latent attribute embeddings in an unsupervised way, 2) application to other attributes like translation domain or target language in multi-lingual systems, 3) optimizing the fine-tuning to affect even less model parameters, 4) an investigation of which attributes are "easier" and "harder" to learn.

Acknowledgements

We would like to thank Anja Austermann from the Google Translate team for discussions about Japanese politeness registers and their classification.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *WMT*.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs. GitHub.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *WMT*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *NAACL*.

Mostafa Elaraby, Ahmed Y. Tawfik, Mahmoud Khaled, Hany Hassan, and Aly Osama. 2018. Gender aware spoken language translation applied to english-arabic. In *ICNLSP*.

Weston Feely, Eva Hasler, and Adrià de Gispert. 2019. Controlling Japanese honorifics in English-to-Japanese neural machine translation. In *ACL WAT*.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. 2020. Flax: A neural network library and ecosystem for JAX. GitHub.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in AI. In *FAT*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *RANLP*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *EMNLP*.

James Kuczmarski and Melvin Johnson. 2018. Gender-aware natural language translation. Technical report, Technical Disclosure Commons.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

Surafel Melaku Lakew, Mattia Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. In *IWSLT*.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In *COLING*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.

Shrimai Prabhumoye, Brendon Boldt, Ruslan Salakhutdinov, and Alan W Black. 2021. Case study: Deontological ethics in NLP. In *NAACL*.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *LREC*.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David Uthus, and Zarana Parekh. 2021. TextSETTR: Label-free text style extraction and tunable targeted restyling. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *NAACL*.

Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, et al. 2018. Mesh-tensorflow: Deep learning for supercomputers. In *NIPS*.

Emmanouil Stergiadis, Satendra Kumar, Fedor Kovalev, and Pavel Levin. 2021. Multi-domain adaptation in neural machine translation through multidimensional tagging. *CoRR*, abs/2102.10160.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *ACL*.

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 *NIPS*.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *COLING WAT*.

A Appendix

A.1 Datasets and baselines

The WMT17 dataset was available via Tensorflow's Datasets. For OpenSubtitles we used a random split to obtain a dev and a test set. Cardinalities of the dev and test set are available in Table 5.

Dataset	Split	Cardinality
WMT17	dev	2999
WMT17	test	3004
OpenSubtitles	dev	4572
OpenSubtitles	test	4566
JESC	dev	2000
JESC	test	2000

Table 5: Cardinalities of datasets.

A.2 Tokenizer configuration

For EN ⇒ DE we trained a joint unigram SentencePiece (Kudo, 2018; Kudo and Richardson, 2018) model with a vocabulary size of 32K. For EN ⇒ JA, following (Feely et al., 2019), we trained two disjoint BPE subword vocabularies with size 32k and character coverage $\alpha = 0.9995$ using the same SentencePiece code.

A.3 SacreBLEU configuration

For German we used the configuration string: BLEU+case.mixed+lang.de-en+numrefs.1+smooth.exp+tok.13a+version.1.4.3; for Japanese we used the configuration string BLEU+case.mixed+lang.ja-en+numrefs.1+smooth.exp+tok.none+version.1.5.1. Note that as in (Feely et al., 2019) the Japanese text was tokenized using the KyTea tokenizer before computing the BLEU score.

A.4 Experiment setup

Our implementation of the Base Transformer is based on the Flax WMT example[8]. On the WMT14 test set, used to verify implementation correctness, our baseline model's and the original Base Transformer's scores (Vaswani et al., 2017) are, respectively, 27.8 BLEU points and 27.3.

We trained on TPUv2 (16 cores) with batch size 256 and used sentence packing (Shazeer et al., 2018) to increase efficiency of accelerator usage. The learning rate was set to 0.0625 with 1k steps of linear warm-up and square-root decay afterwards. We used the default Adam optimizer and a dropout rate of 0.1. For EN ⇒ DE we trained for a minimum of 100k steps and after that used early stopping, evaluating every 10k steps, on the dev's set BLEU score with a patience of 5; results were evaluated on the best checkpoint for the dev set. For EN ⇒ JA we used a patience of 10 and we used two separate embeddings on top of the separate BPE vocabularies following the configuration reported in (Feely et al., 2019). For EN ⇒ DE we used beam search with beam size 4 and length-penalty 0.6. For EN ⇒ JA we used beam search with beam size 10 and length-penalty 0.9; these parameters having been fine-tuned for the Base on the dev set.

We were unable to replicate the performance score of 18.8 for the Base model in (Feely et al., 2019) even though the improvements we saw for controlling politeness are consistent with their results. We conjecture these might be due to a mismatch of some model configuration or to a different setup for evaluating the BLEU score.

[8] https://github.com/google/flax/tree/master/examples/wmt

A.5 Tagging configuration

For length (resp. monotonicity) we used 5 (resp. 3) buckets whose boundaries were chosen so that each bucket contains approximately the same amount of data. For the tagging models that have a Neutral mode, this was simulated by a "neutral" masking tag that replaces each original tag independently with a 20% probability. When using tags interventions were made by shifting, i.e. shifting each tag id by k positions and clipping to a valid tag; so if there are l tags there are $2 * l - 1$ possible interventions where $k = 0$ corresponds to the Oracle mode.

A.6 BLEU scores for the different permutations of tagging

In Tables 6 and 7 we report the BLEU scores on WMT and OpenSubtitles for models trained with the different permutations of the tags. The best and worst results are indicated by an asterisk (*) and are reported in the main paper.

A.7 Annotation of German politeness

We used the rules from (Sennrich et al., 2016), that look at the German 2nd person pronouns 'Sie' (*polite* 'you') and 'du' (*informal* 'you') and the corresponding verbs. Here the parser is mainly used to correctly classify ambiguous pronouns, e.g. "ihr"

Model	Mode	BLEU
$\text{Tag}(\text{L}, \text{M}, \text{P}_d)$	Oracle	27.32*
$\text{Tag}(\text{L}, \text{P}_d, \text{M})$	Oracle	26.58*
$\text{Tag}(\text{M}, \text{L}, \text{P}_d)$	Oracle	26.62
$\text{Tag}(\text{M}, \text{P}_d, \text{L})$	Oracle	27.06
$\text{Tag}(\text{P}_d, \text{L}, \text{M})$	Oracle	27.13
$\text{Tag}(\text{P}_d, \text{M}, \text{L})$	Oracle	27.28

Table 6: BLEU scores on WMT EN-DE for the different permutations of tags. * difference is statistically significant with $pval < 10^{-10}$.

Model	Mode	all	*unknown*	*polite*	*informal*
$\text{Tag}(\text{L}, \text{M}, \text{P}_d)$	Oracle	21.99	22.05	24.94	20.47
$\text{Tag}(\text{L}, \text{P}_d, \text{M})$	Oracle	21.17	21.14	23.66	20.21
$\text{Tag}(\text{M}, \text{L}, \text{P}_d)$	Oracle	21.50	21.41	23.78	20.69
$\text{Tag}(\text{M}, \text{P}_d, \text{L})$	Oracle	21.60	21.65	24.06	20.34
$\text{Tag}(\text{P}_d, \text{L}, \text{M})$	Oracle	21.73	21.81	23.35	20.67
$\text{Tag}(\text{P}_d, \text{M}, \text{L})$	Oracle	21.81	21.80	24.59	20.59

Table 7: BLEU scores of a WMT-trained model on OpenSubtitles by politeness split for all permutations of tags. Sizes: all: 4566, unknown: 3617, polite: 276, informal: 673.

to make sure it refers to a second person. In Table 8 we report relative frequencies of the data annotated as *polite* or *informal*.

A.8 Classification accuracy on the politeness rewriting task

We took the test subsets of OpenSubtitles where the references is classified as *polite* or *informal* and translate the source side into either *polite* or *informal* mode and run the rule-base classifier on the translations to find out the realized rewriting accuracy (Table 9). Thanks to the flexibility of the additive approach, we were able to match this accuracy by fine-tuning the *informal* multiplier for *after training*. For example, for the model $\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$ the multiplier value for *informal* that we found by grid-search was 1.9 which resulted in a rewriting accuracy of 79.6% in Oracle mode and resp. 80.4%. In terms of BLEU scores this translates to respective improvements of 19.68 and 20.22. In our grid-search we optimized for the BLEU score; however there is a trade-off with the rewriting accuracy as the latter can be further increased above 85% while keeping the BLEU score above 18.0.

Dataset	*informal*	*polite*
WMT	1.2%	7.9%
OpenSubtitles	15.3%	6.2%

Table 8: Relative frequency of politeness annotation for German.

Model	*polite*	*informal*
Base	62.7	8.5
$\text{Tag}(\text{L}, \text{M}, \text{P}_d)$	87.1	82.8
$\text{Tag}(\text{L}, \text{P}_d, \text{M})$	85.9	79.0
$\text{Tag}(\text{M}, \text{L}, \text{P}_d)$	85.4	78.1
$\text{Tag}(\text{M}, \text{P}_d, \text{L})$	87.2	81.0
$\text{Tag}(\text{P}_d, \text{L}, \text{M})$	86.6	81.7
$\text{Tag}(\text{P}_d, \text{M}, \text{L})$	87.7	81.8
$\text{Tag}_{\text{mask}}(\text{L}, \text{M}, \text{P}_d)$	84.4	79.5
$\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$	85.7	79.9
$\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$	77.8	70.2
$\text{Add}_2(\text{L}, \text{M}_{0.1}, \text{P}_c)$	77.9	70.8

Table 9: Classification accuracy (%) on rewriting into *polite* and *informal* for the OpenSubtitles test set.

Formality Level	Multipliers
unknown	0.5
informal	1.0
polite	1.5
formal	2.0

Table 10: Multipliers for Japanese politeness.

A.9 Annotation of Japanese politeness

For Japanese a politeness and formality registers can be inferred from verb endings and presence of honorific expressions. We took the rules from Table 3 of (Feely et al., 2019) and used the SpaCy parser. In Listing 1 we report the code we used for annotation. The `formal_verbs`, `polite_verbs` and `informal_verbs` are Python's sets of strings that we report in Tables 16 and 17. Each string represents the way SpaCy parses a grammatical rule of politeness inside a sentence and for each string we report how a full example sentence was parsed by SpaCy. The values of the multipliers used for the continuous feature are in Table 10.

A.10 Quantifying non-monotonicity

To evaluate translation monotonicity one would like to measure how the change in the monotonicity,

$\delta(s)$, is affected when requesting translations with lower or higher $\delta(s)$. First note that $\delta(s)$ is already normalized to lie in $[0, 1]$ as the ratios $i/n, j/m$ in its definition rescale the sentence lengths to the unit interval. In the limit case of $n, m \to \infty$ we might think of of an alignment, which concretely consists of pairs $(i/n, j/m)$, as representing a continuous curve $t \to c(t)$ mapping $[0, 1]$ to $[0, 1]$. Now $\delta(s)$ would become the L^1-distance between c and the identity mapping $t \to t$. In the finite case, if we think of an alignment as a curve, possibly with jumps, we can then think of reparametrizing it to be defined on a domain corresponding to the sentence length; we thus propose to multiply $\delta(s)$ by the length of the translation s_{target}, to arrive at interpretation of $\text{len}(s_{\text{target}}) \times \delta(s)$ as a non-monotonicity measure – by how many positions in the translation tokens deviate from the corresponding tokens in the input sentence. Now, given a set of translations S we define the degree of their non-monotonicity as:

$$\Delta(S) = \sum_{s \in S} \text{len}(s_{\text{target}}) \times \delta(s),$$

which quantifies by how many token positions the translations cumulatively deviate from the corresponding input sentences.

However, we are interested in comparing monotonicity between sets of translations; so given two sets S, S' of translations of the same inputs we look at $\Delta(S)/\Delta(S')$. This alone, however, would give a partial picture as it does not take into account the distribution of the $\delta(s)$. Therefore, we propose to slice $\Delta(S)$ at cuts by looking at subsets of S, S' where $\delta(s) \geq \text{cut}$. Put together, we define the relative non-monotonicity as:

$$\Delta_{\text{rel}}(\text{cut}) = \frac{\Delta(t : \text{translation with } \delta(t) \geq \text{cut})}{\Delta(t : \text{reference with } \delta(t) \geq \text{cut})},$$

which compares the translations with the references, with values larger than 1.0 indicating more re-orderings than the references and vice-versa.

A.11 Decreasing monotonicity

When asking the model to *decrease* monotonicity, we observed that the continuous approach gives a more fine-grained control. For example in Figure 3 we compare a tagging and a continuous model in the direction EN ⇒ DE for different values of the interventions. Note that asking to reduce monotonicity does result in lower BLEU scores, so to make a fair comparison with tagging we fixed a range of values for the continuous interventions that does not lead to a worse reduction in BLEU than tagging. Here we observe that with the continuous feature we have a smoother and broader range of possible effects. For Japanese, besides a similar situation, we also found a significant difference between the oracle mode for the continuous and the tagging approaches. In oracle mode, we would expect the translations to closely match the references and hence the $\Delta_{\text{rel}}(\text{cut})$ to stay close to the ideal line $y = 1.0$ as cut varies. In Figure 4 we see that at a certain point the continuous approach performs better than tagging; for example at cut $= 0.3$ the tagging model has already increase around $y = 1.5$ while the continuous approach is still around $y = 1.07$. Note there we are not yet at the tail of the distribution as for the references the $\Delta(0.3)/\Delta(0)$ is at about 15%.

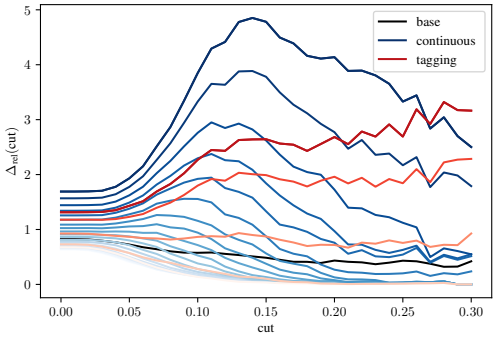


Figure 3: How we can vary monotonicity EN ⇒ DE for different interventions. The model using tagging is $\text{Tag}(\text{L}, \text{M}, \text{P}_d)$ while the continuous model is $\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$. Darker shades of the same color represent larger values of the intervention δ.

A.12 Fine-tuning results

In Table 11 we report the BLEU scores on WMT17 and the *formal/informal* splits of OpenSubtitles for the selected checkpoints. On WMT we still see good performance with similar scores between Neutral and Oracle mode. The results on OpenSubtitles show that the model learns to use the politeness annotation to improve the quality of translations.

A.13 Exemplars

In Table 12 there is an example of varying the length of a translation for German. Here the controllable model is not simply dropping tokens from

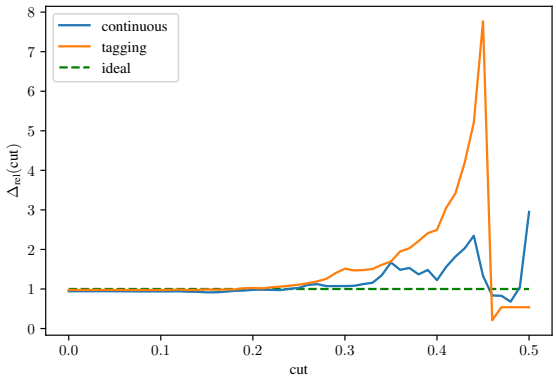


Figure 4: Comparing continuous and tagging for EN $\Rightarrow$ JA monotonicity in oracle mode. The model using tagging is $\text{Tag}_{\text{inv}}(\text{L}, \text{M}, \text{P}_d)$ while the continuous model is $\text{Add}(\text{L}, \text{M}_{0.1}, \text{P}_c)$.

the end even in the range of i_r where we found it comparable to the rewriter in terms of BLEU score. For example at $i_r = 0.6$ it takes out some additional information like the year of the restoration but keeps the main verb. Note that in neutral mode the translation was shorter than the reference and for $i_r = 1.0$, corresponding to oracle mode, the system tries to match the length of the translation. In Table 13 we consider an example of varying the length of a translation in Japanese. Going from shorter to longer translations: the system first translates the main verb/imperative ($i_r = 0.3$), then it translates the "together" ($i_r = 0.5$) and keeps refining the verb ending till $i_r = 1.0$; after that length is increased by introducing explicitly personal pronouns or the "why?" that would be optional in Japanese. As a side effect, length interventions are generating also a broader grammatical variety of translations.

In Table 14 we have some exemplars for monotonicity control. In the first German example the reference is less monotonic because the subject comes at the end and the information about the 58-years old is first; the more monotonic translation corrects the order. In the first Japanese example to increase monotonicity the model adds the personal pronoun "I" that is missing from the reference, shifting the alignment. In the second Japanese example we observed that setting the target δ for $\delta(s)$ small produces a bit more variety of translations (an advantage of a continuous representation of monotonicity) where the model tries to get a translation where the time information about "a few years" comes towards the end of the sentence.

	Model	Mode	Dataset	BLEU
Reset	15%	Neutral	WMT	26.78
		Oracle	WMT	26.55
		Neutral	OS-informal	13.32
		Oracle	OS-informal	17.55
		Neutral	OS-polite	22.17
		Oracle	OS-polite	23.11
	35%	Neutral	WMT	26.74
		Oracle	WMT	26.53
		Neutral	OS-informal	13.37
		Oracle	OS-informal	18.24
		Neutral	OS-polite	22.35
		Oracle	OS-polite	23.12
Fine Tune	20%	Neutral	WMT	26.66
		Oracle	WMT	26.87
		Neutral	OS-informal	12.96
		Oracle	OS-informal	18.04
		Neutral	OS-polite	22.81
		Oracle	OS-polite	23.24
	35%	Neutral	WMT	26.73
		Oracle	WMT	26.90
		Neutral	OS-informal	13.30
		Oracle	OS-informal	19.23
		Neutral	OS-polite	22.96
		Oracle	OS-polite	23.79

Table 11: BLEU scores for fine-tuning.

In Table 15 we show how the politeness Oracle helps in German and Japanese to get a translation more close to the reference since the English input sentences admit different translations in the target languages, e.g. regarding choices for the informal/-formal pronouns for German, or verb endings and honorifics for Japanese.

A.14 Model implementation

In Listing 2 we give an indication of how the model can be implemented in Flax. Note that for simplicity we assume that the encoder and the two parts of the decoder are already implemented, e.g. by taking them from the WMT example in the Flax library. To make the code listing clear and short we assume the each row of the batch contains a single sentence, i.e. that the model is not implemented to work with sentence packing. In the case of sentence packing a few modifications are necessary but are easy to implement using either `jax.lax.scan` or `jnp.einsum`, depending on how one keeps track of the sentence id.

input	Gee and Schwartzman are scheduled to discuss and share images of the 1926 Bertram Goodhue design as well as the 1993 restoration and addition by Hardy Holzman Pfeiffer Associates.
reference	Es ist vorgesehen, dass Gee und Schwartmann Bilder des 1926 Bertram Goodhue Designs sowie die 1993 Restoration und Ergänzung von Hardy Holzman Pfeiffer Associates diskutieren und teilen werden.
neutral	Gee und Schwartzman sollen die Bilder des Bertram Goodhue-Designs von 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer Associates von 1993 diskutieren und teilen.
$i_r = 0.6$	Gee und Schwartzman sollen Bilder des Bertram Goodhue Designs von 1926 sowie die Restaurierung und Ergänzung diskutieren.
$i_r = 0.7$	Gee und Schwartzman sollen Bilder des Bertram Goodhue Designs 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer.
$i_r = 0.8$	Gee und Schwartzman sollen die Bilder des Bertram Goodhue Designs 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer diskutieren.
$i_r = 0.9$	Gee und Schwartzman sollen die Bilder des Bertram Goodhue-Designs 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer Associates 1993 diskutieren.
$i_r = 1.0$	Gee und Schwartzman sollen die Bilder des Bertram Goodhue-Designs 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer Associates aus dem Jahr 1993 diskutieren und teilen.
$i_r = 1.10$	Gee und Schwartzman sollen die Bilder des Bertram Goodhue-Designs von 1926 sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer Associates aus dem Jahr 1993 diskutieren und mit ihnen teilen.
$i_r = 1.20$	Gee und Schwartzman sollen die Bilder des Bertram Goodhue-Designs von 1926, sowie die Restaurierung und Ergänzung von Hardy Holzman Pfeiffer Associates aus dem Jahr 1993 diskutieren und mit ihnen in Verbindung bringen.

Table 12: Modifying length of a German translation.

input	why don't you come sit down with me?
reference	こっちに来て一緒に座らない?
	over here come together not sitting down?
$i_r = 0.30$	座って
	sit down!
$i_r = 0.50$	一緒に 座って
	together sit down!
$i_r = 0.70$	一緒に座ろう
	together let's sit down
$i_r = 0.90$	一緒に 座ったら?
	together why don't sit down?
$i_r = 1.00$	一緒に座らないか?
	together not sitting down?
$i_r = 1.10$	俺と一緒に座ったらどうだ?
	with me together when sitting down how's it?
$i_r = 1.20$	なぜ私と一緒に座らない?
	why with me together not sitting down?
$i_r = 1.30$	なぜ私と一緒に座らないの?
	why with me together not sitting down?
$i_r = 1.50$	なぜあなたは私と一緒に 座らないか?
	why you with me together not sitting down?

Table 13: Modifying length of a Japanese translation.

input	It was a first for the 58-year-old.
reference	Für den 58-Jährigen war es eine Premiere.
make more monotone	Es war ein erster für den 58-Jährigen.
match reference monotonicity	Für den 58-Jährigen war dies eine erste.
input	i've already met four people
reference	既に4人会ったわ already 4 people met
make more monotone	私は既に4人に 会った I already 4 people met
match reference monotonicity	すでに4人に会っている already 4 people meeting
input	this thing was discovered just a few years
reference	これはたった数 年前に発見されたもので this only a number years before discovered conj. particle
make more monotone	発見されたのはほんの 2~3年前 discovered subj. particle only 2~3 years before
make more monotone	この発見はほんの数年 で this discovery already number of years prep. since
make more monotone	これを発見したのは わずか数年で this discovered subj. particle a little number years prep. since
match reference monotonicity	これはほんの 数年前に発見されたもので this already a number years before discovered conj. particle

Table 14: Increasing monotonicity.

input	You’re good, aren’t you?
reference	Dir geht’s gut, nicht?
neutral	Sie sind gut, sind Sie nicht gut?
match reference politeness	Du bist gut, bist du nicht?
input	why don’t we catch that plane?
reference	飛行機を捕まえましょう plane catch let’s (verb ending)
neutral	飛行機はどう？ plane how about?
match reference politeness	飛行機を捕まえましょう plane catch let’s (verb ending)
input	good morning, okazaki.
reference	おはようございます、岡崎さん good morning hon. ozazaki hon.
neutral	おはよう、岡崎 good morning ozazaki
match reference politeness	おはようございます、岡崎さん good morning hon. ozazaki hon.
input	who are you? tell me your name.
reference	一体何 者 だ 名前を 言え what the heck person are name tell
neutral	あなたは誰ですか? you who/person are?
match reference politeness	誰だ? 名前を言え who/person are? name tell

Table 15: Matching politeness of references.

Rule	Strings in Rule	Example
informal_verbs	だ AUX	夏休み NOUN は ADP 明日 NOUN から ADP だ AUX
	だっ AUX た AUX	夏休み NOUN は ADP 昨日 NOUN から ADP だっ AUX た AUX
	じゃ AUX ない ADJ	彼女 PRON たち NOUN が ADP 有名 ADJ じゃ AUX ない ADJ
	じゃ AUX なかっ ADJ た AUX	彼女 PRON たち NOUN が ADP 有名 ADJ じゃ AUX なかっ ADJ た AUX
	だろう AUX	雨 NOUN が ADP 降る VERB だろう AUX
	だ AUX から SCONJ	だ AUX から SCONJ なん PRON だ AUX
	だ AUX けど SCONJ	悪い ADJ ん SCONJ だ AUX けど SCONJ
	だ AUX って PART	だ AUX って PART 熱く ADJ ない ADJ だ AUX
	だ AUX っけ	だ AUX っけ PART 熱く ADJ ない ADJ だ AUX
	そう ADV だ AUX	はい INTJ そう ADV だ AUX
	よう AUX だ AUX	はい INTJ よう AUX だ AUX
polite_verbs	です AUX	女の子 NOUN です AUX
	でし AUX た AUX	昨日 NOUN は ADP あなた PRON の ADP 誕生 NOUN 日 NOUN でし AUX た AUX
	ない AUX	今日 NOUN は ADP 曇り VERB く AUX ない AUX
	なかっ ADJ た AUX	一昨 NOUN 日 NOUN は ADP 曇り NOUN なかっ ADJ た AUX
	ます AUX	お NOUN 弁当 NOUN を ADP 買い VERB ます AUX
	まし AUX た AUX	お NOUN 弁当 NOUN を ADP 買い VERB まし AUX た AUX
	ませ AUX ん AUX	お NOUN 弁当 NOUN を ADP 買い VERB ませ AUX ん AUX
	ましょう AUX	お NOUN 弁当 NOUN を ADP 買い VERB ましょう AUX
	でしょう AUX	雨 NOUN が ADP 降る VERB でしょう AUX
	ください VERB	ビール NOUN を ADP ください VERB
	なさい AUX	この DET 猫 NOUN は ADP 見 VERB なさい AUX
	で AUX ある AUX	先生 NOUN で AUX ある AUX
	で AUX わ PART ない ADJ	先生 NOUN で AUX わ PART ない ADJ

Table 16: Annotation of Politeness for Japanese (I). Components of each rule and an example sentence.

Rule	Strings in Rule	Example
formal_verbs	ござい AUX ます AUX	駅 NOUN が ADP そちら PRON で AUX ござい AUX ます AUX
	いらっしゃい VERB ます AUX	父 NOUN は ADP 銀行 NOUN に ADP いらっしゃい VERB ます AUX
	あり AUX ます AUX	猫 NOUN は ADP 五 NUM 匹 NOUN あり AUX ます AUX
	なさい AUX ます AUX	この DET 猫 NOUN は ADP 見 VERB なさい AUX ます AUX
	致し VERB ます AUX	宿題 NOUN を ADP 致し VERB ます AUX
	ご覧 NOUN に ADP なり VERB ます AUX	彼 PRON は ADP 映画 NOUN で ADP 映画 NOUN を ADP ご覧 NOUN に ADP なり VERB ます AUX
	拝見 VERB し AUX ます AUX	うち NOUN は ADP 山 NOUN を ADP 拝見 VERB し AUX ます AUX
	お NOUN 目 NOUN に ADP 掛かり VERB ます AUX	明日 NOUN は ADP お NOUN 目 NOUN に ADP 掛かり VERB ます AUX
	お NOUN いで VERB に ADP なり VERB ます AUX	兄 NOUN は ADP お NOUN いで VERB に ADP なり VERB ます AUX
	伺い VERB ます AUX	明日 NOUN 朝 NOUN イチ PROPN で ADP 伺い VERB ます AUX
	参り VERB ます AUX	お NOUN 店 NOUN へ ADP 参り VERB ます AUX
	存知 NOUN し AUX ます AUX	何 PRON 存知 NOUN し AUX ます AUX
	存じ VERB 上げ AUX ます AUX	何 PRON 存じ VERB 上げ AUX ます AUX か PAR
	召し上がり VERB ます AUX	酒 NOUN を ADP 召し上がり VERB ます AUX
	頂く VERB	仕事 NOUN は ADP 頂く VERB
	頂き VERB ます AUX	仕事 NOUN は ADP 頂き VERB ます AUX
	頂い VERB て SCONJ	大学 NOUN 生 NOUN は ADP 本 NOUN を ADP 頂い VERB て SCONJ い AUX ます AUX
	差し VERB あげ AUX ます AUX	靴 NOUN は ADP 差し VERB あげ AUX ます AUX
	下さい VERB ます AUX	先生 NOUN は ADP お NOUN ちゃ NOUN を ADP 下さい VERB ます AUX
	おっしゃい VERB ます AUX	先生 NOUN は ADP おっしゃい VERB ます AUX
	申し VERB 上げ AUX ます AUX	ご NOUN 挨拶 NOUN を ADP 申し VERB 上げ AUX ます AUX

Table 17: Annotation of Politeness for Japanese (II). Components of each rule and an example sentence.

Listing 1: Annotation of Japanese politeness

```
import spacy
# Initialize SpaCy.
nlp_jp = spacy.load(
        'ja_core_news_sm')

def annotate(jp_txt: str):
  jp_doc = nlp_jp(jp_txt)
  jp_doc = expand_doc(jp_doc)
  politeness = '<unknown>'
  # formal > polite > informal
  done = False
  for table, tag in zip(
    (formal_verbs,
    polite_verbs, informal_verbs),
    ('<formal>', '<polite>',
    '<informal>')):

    for form in table:
      if form in jp_doc:
        politeness = tag
        done = True
        break
        if done:
          break
  return politeness
```

Listing 2: Implementation of the model

```
import flax
import flax.linen as nn

class Model(nn.Module):
  encoder: nn.Module
  # First few layers of decoder
  decoder_lo: nn.Module
  # Last N layers of decoder
  decoder_up: nn.Module
  # Cardinality of attributes
  nr_attr: int
  emb_dim: int

  @nn.compact
  def __call__(inputs, targets,
    attr_id, attr_weight):
    # attr_id: B, nr_attr
    # attr_weight: B, nr_attr
    # inputs, targets: B, T

    # Construct vector V
    V = nn.Embed(
      num_embeddings=nr_attr,
      features=emb_dim)(attr_id)
    attr_weight = jnp.expand_dims(
       attr_weight, axis=-1)
    V = V * att_weight
    # sum over the # of attributes
    # so shape B, emb_dim
    V = jnp.sum(V, axis=1)
    # add dimension to sum
    # across time steps
    V = jnp.expand_dims(V, axis=1)

    encoded = self.encoder(
        inputs)
    dec1 = self.decoder_lo(
        targets, encoded)
    logits = self.decoder_up(
        dec1, encoded + V)
    return logits
```