Soft Alignment Objectives for Robust Adaptation in Machine Translation

Anonymous ACL submission

Abstract

Domain adaptation allows generative language models to address specific flaws caused by the domain shift of their application. However, the traditional adaptation by further training on in-domain data rapidly weakens the model's ability to generalize to other domains, making the open-ended deployments of the adapted models prone to errors. This work introduces novel training objectives built upon a semantic similarity of the predicted tokens to the reference.

Our results show that (1) avoiding the common assumption of a single correct prediction by constructing the training target from tokens' semantic similarity can mitigate catastrophic forgetting during domain adaptation, while (2) preserving the quality of adaptation, (3) with negligible additions to compute costs. In the broader perspective, the objectives grounded in a soft token alignment pioneer the exploration of the middle ground between the efficient but naive exact-match token-level objectives and expressive but computationally- and resource-intensive sequential objectives.

1 Introduction

Large language models (LLMs) based on instances of encoder-decoder architecture (Neyshabur et al., 2015) nowadays serve as a strong default in generative applications of NLP, such as summarization or machine translation, mainly thanks to their outstanding ability to fluently model language. These models still face issues with *adequacy* of the generated text (Ustaszewski, 2019) when applied to a domain of data that differ from the training domain, but such errors can be mitigated using domain adaptation (Saunders, 2021).

Identically to the pre-training of the generative LLMs, the adaptation is commonly carried out using Maximum Likelihood Estimation (*MLE*) objective with teacher forcing (Bahdanau et al., 2015). The widespread of such approach might be dedicated to its data and resource efficiency.

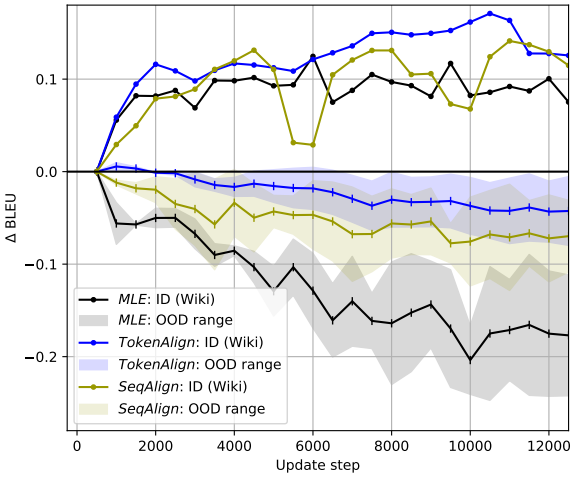

Figure 1: In-domain (ID) and out-of-domain (OOD) change of the original BLEU in domain adaptation of a translation model using ***MLE*** and the two introduced objectives: ***TokenAlign*** and ***SeqAlign***. Adaptation of Transformer-base model on Wikipedia, evaluated on a held-out set of the adapted domain (in-domain, ID) and a variety of out-of-domain (OOD) datasets (§4.2).

Despite these benefits, model adaptation using *MLE* notoriously comes for a price of over-specialisation to the target domain, also referred to as *catastrophic forgetting* (Goodfellow et al., 2014), characterized by a continuous decay of model performance on the inputs from the domains *other* than the adaptation domain (see Figure 1).

Our work addresses the loss of robustness characteristic for domain adaptation by extending the *MLE* objective with complementary objectives. We construct targets of these objectives through soft alignment of model predictions to the reference and quantify the instantaneous quality of model outputs by the quality of such alignment.

In our experiments, we find that using such objectives in domain adaptation can address the loss of model robustness, eliminating a major portion of model performance loss on out-of-domain (OOD), caused by conventional adaptation while reach-

ing comparable or higher qualitative gains on the adapted domain.

The main contributions of our work are the following. (i) We present a framework for training generative language models with an alternative training signal based on token similarity provided by an arbitrary embedding model. A similar methodology can be applied for robust training and adaptation of any language model. (ii) We introduce efficient and accurate training objectives that alleviate catastrophic forgetting of domain adaptation in NMT without losing adaptation quality. (iii) We study the aspects that impact LLMs' robustness, relevant for the training and fine-tuning of any generative LLM. Among others, we find that a more robust model can be obtained merely by exposing a generative model to its own predictions during the training.

This paper is structured as follows. Section 2 surveys and compares our work to the existing work in training and adapting robust generative LLMs. Section 3 introduces two main objectives that we experiment with: *TokenAlign* and *SeqAlign*. Section 4 describes our experimental methodology and ablation analyses and Section 5 summarizes our findings, highlighting the broader implications.

2 Background

Language generation is the modus operandi for a set of problems requiring open-ended sequence of tokens as the answer. Machine translation is the representative of this group that we focus on, but other tasks such as summarization (Lewis et al., 2020), vision captioning (Wang et al., 2022), or more recently prompting (Carlsson et al., 2022) are also applications of the described framework.

In the commonly-used auto-regressive settings, for each encoded input X_j and reference Y_j, a *language model* Θ: $\Theta(X_j, Y_{j,1..i-1}) \rightarrow \mathbb{R}^{|vocab|}$ is trained to generate a sequence by maximising the probability of generating the i-th token $y_{ji} = \arg\max(\Theta(X_j, Y_{j,1..i-1}))$ *matching* the reference Y_{ji}, while minimising the probability of the other tokens of the vocabulary, as conditioned by the *previous* reference tokens $Y_{j,1..i-1}$:

$$\max p(y_{ji} = Y_{ji} | Y_{j,1..i-1}, X_j, \Theta) \qquad (1)$$

This objective is implemented in the commonly-used Maximum Likelihood Estimation (*MLE*) objective, that minimises a cross-entropy (CE) of predicted distribution of $\Theta(X_j, Y_{j,1..i-1})$ to the *expected* distribution, which is a one-hot encoding E_{ji} of the *true* reference token Y_{ji} over the model vocabulary, on the position i:

$$\mathcal{L}_{MLE}(\Theta) = \min\left(-\log \frac{\exp(\Theta(X_j, Y_{j,1..i-1}))}{\exp(E_{ji})}\right) \qquad (2)$$

This objective is commonly used both for training (Bahdanau et al., 2016; Vaswani et al., 2017) and adaptation (Servan et al., 2016; Saunders, 2021) of generative LLMs.

While the adaptation brings benefits in modeling domain-specific terminology (Sato et al., 2020) or in avoiding inadequate generation artifacts such as repetitions or hallucinations (Etchegoyhen et al., 2018), it comes for a price of model generalization, known also as catastrophic forgetting (Fig. 1). The adapted models improve on the adapted domain but gradually perform worse on other domains.

Selected work in domain adaptation of MT also addresses the mitigation of catastrophic forgetting. Freitag and Al-Onaizan (2016) obtain more robust model by ensembling the original model with the adapted one. Thompson et al. (2019) regularize the training using Fischer Information Matrix. Chu et al. (2017) enhance model robustness with mixing the pre-training and adaptation samples. More similar to ours, Dakwale and Monz (2017) use regularization of the loss based on the distillation. Our work differs from this branch in both data and computational requirements. We do not presume availability of pre-training data, nor do we need to perform the simultaneous inference with the original models.

Specific problem of *MLE* and other approaches is referred to as *exposure bias*: while in the teacher-forced training, the model's i-th prediction $\Theta(X_j)_i$ is conditioned by the correctly-generated previous tokens from the reference $Y_{j,1..i-1}$, in generation, the model conditions its predictions on its *own* outputs $\Theta(X_j)_{1..i-1}$. This discrepancy might be magnified under a domain shift where the model does *not learn* to *follow* reference in generation.

Exposure bias can be addressed by sampling strategies constructing the sequence of previous tokens $Y_{j,1..i-1}$ by sampling from both reference and generated tokens (Bengio et al., 2015; Zhang et al., 2019), but such mixed priors do not always persist the original meaning. Different work utilize *sequential objectives*, such as Minimum Risk Training (MRT) (Ranzato et al., 2016) that

optimize model weights based on a complete output sequence, regardless of specific tokens. Such evaluation is provided by one of the MT measures (Shen et al., 2016; Wang and Sennrich, 2020; Unanue et al., 2021) or by a feedback of adversarial model, penalizing Θ, for instance, for distinguishing generated and original text (Yang et al., 2018; Yu et al., 2016) or violating language morphology (Mi et al., 2020). Despite some gains, sequence-level objectives face specific problems of reinforcement learning (RL), such as a fragility to the optimization settings (Pineau et al., 2021), and are also more resource-demanding as they require a sequence of predictions for a single update, which constrain their applicability in low-resource domain adaptation. Additionally, further analyses of Choshen et al. (2020) show that sequential objectives reach performance gains comparable to a constant training signal, raising doubts about the justification of their extensive data and compute demands. Inspired by this finding, we also critically assess our methods against a random feedback baseline (§4.3).

Closer to our work, others construct the training signal from *alignment* of model's instantaneously generated sequence to the reference. Xu et al. (2019) build soft alignment between fully-generated hypotheses based on hidden states of bidirectional LSTM encoder-decoder and weigh the predicted probability distribution by such alignment in the training objective. Similarly, Lu et al. (2020) complement *MLE* and sentence-level objective with the objective minimizing a dot-product of the best-matching hidden representations of tokens of a hypothesis and a reference. Chen et al. (2019) and later Zhang et al. (2020a) introduce the matching scheme that use the Optimal transport cost (Kusner et al., 2015) of the embeddings of reference to the hypothesis as their objective loss. All of these studies use instances of recurrent encoder-decoder networks and hidden encoder representations as to the token embeddings.

Our work extends the branch of research utilizing token representations in the training but differs in some important aspects; We focus on more challenging settings of very-low to medium-resource adaptation, instead using more recent Transformer models pre-trained on a large mixture of domains (Tiedemann and Thottingal, 2020). Additionally, instead of building the alignment on the trained model embeddings, our framework uses static pre-trained embeddings as token representations that remain domain-agnostic in adaptation.

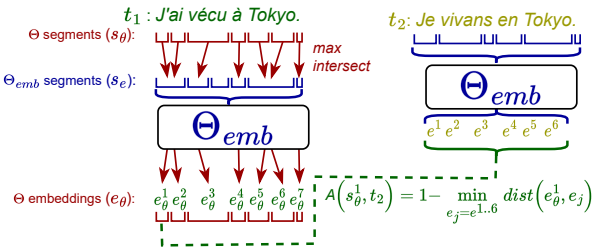

Figure 2: **Token alignment mechanism** allows us to represent tokens (s_θ) of the trained model Θ using an arbitrary Embedding model Θ_{emb}. We define *alignment* of a Θ's segment e^i_θ to another text t_2 through a distance of their embeddings given by Θ_{emb}.

3 Soft Alignment Objectives

Following section introduces two novel objectives that use the described alignment mechanism as their target.

3.1 Token Alignment

Unlike the previous work (Xu et al., 2019; Lu et al., 2020; Chen et al., 2019), our alignment circumvents the representation using model's own embeddings, as we argue that model's own feedback in adaptation is likely impacted by the forgetting.

The alignment mechanism is overviewed in Figure 2. As the vocabulary of our chosen embedding model Θ_{emb} is usually not aligned with the vocabulary of the trained model Θ, we first tokenize input text t_1 using both Θ and Θ_{emb}'s tokenizer, obtaining segments s_θ and s_e respectively. We *match* each segment s^i_θ with a segment s^j_e of Θ_{emb} such that s^j_e has the largest spatial overlap with s^i_θ. Therefore, each Θ's segment s^i_θ gets associated with an embedding of Θ_{emb}.

Subsequently, we define an *alignment* $\mathcal{A}$ of any segment s^i_θ to another text t_2:

$$\mathcal{A}(s^i_\theta, t_2) = 1 - \min_{e^j \in \Theta_{emb}(t_2)} dist(e^i_\theta, e^j) \qquad (3)$$

where *dist* is a distance measure defined for the selected embedding system. In our experiments, we use standard Euclidean distance as the measure. A more explicit description of the alignment algorithm can be found in Appendix D.

3.2 *TokenAlign* Objective

TokenAlign is designed as a minimal adjustment to *MLE* (Eq. (2)), inheriting most of its efficiency.

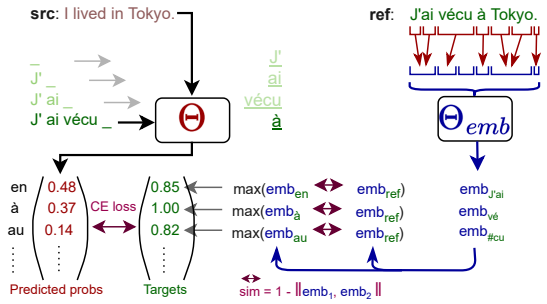

Figure 3: ***TokenAlign* objective** replaces one-hot targets of *MLE* with *alignment* $\mathcal{A}$ (§3.1) computed as a maximum similarity between the embeddings of the candidate and reference tokens, guiding the model Θ to predict higher probabilities for the tokens *similar* to the reference, according to the representations of Θ_{emb}.

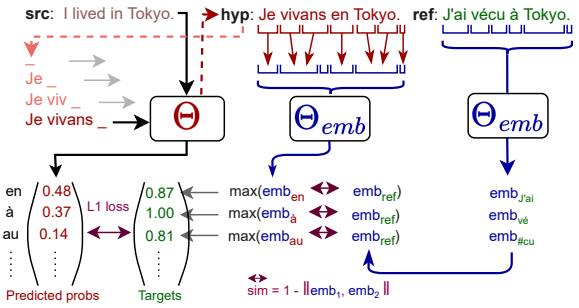

Figure 4: ***SeqAlign* objective** further replaces the reference prefixes in the training with Θ's own predictions. Using the token alignment scheme $\mathcal{A}$ (§3.1), we differentiate the quality of the predicted tokens as their *best possible match* to the reference, according to the aligned tokens' embeddings.

However, *TokenAlign* circumvents the naive assumption of *MLE* that only a single token of the reference is a correct prediction by also encouraging the model to up-weight predictions that can be accurately aligned to the reference (Fig. 3):

$$\mathcal{L}_{TAlign}(\Theta) = \min\left(-\log \frac{\exp(\Theta(X_j, Y_{j,1..i-1}))}{\exp(\mathcal{A}(voc_\theta, Y_j))}\right) \quad (4)$$

where voc_θ is the token vocabulary of Θ, and $\mathcal{A}(s_\theta^{1..|\theta|}, Y_j)$ are the *alignments* for each token of the vocabulary (s_θ^i) to the given reference Y_j.

Relying on the same training approach as with the conventional *MLE* objective, *TokenAlign* presents a alternative of the *MLE* of similar data and compute efficiency (Appendix C). However, *TokenAlign* still does not address the exposure bias as the model Θ is still updated conditionally to the previous *reference* tokens $Y_{1..i-1}$ as the prefixes, rather than its own outputs.

3.3 *SeqAlign* Objective

By utilizing the token-level embeddings, we circumvent the feedback sparsity of conventional sequence-level objectives and provide the language model with updates for every prediction step, rather than its whole hypothesis.

Hence, instead of constructing the prediction prefixes from the references Y, we construct the prefixes by iteratively selecting the tokens according to the current outputs of Θ; Specificaly, we use Θ's outputs as a *probability distribution* and construct a generation strategy Π^θ that stochastically *samples* next token(s) from this distribution.

Consequentially, instead of generating a *single* hypothesis for each input, we can obtain a *set* of hypotheses $\hat{Y}_j \sim \Pi^\theta(X_j, \Theta)$ that can be aligned to Y_j and used by *SeqAlign* to condition the updates of Θ (Fig. 4). A desirable property of this approach is that the prefixes of such hypotheses are realistically likely to occur during Θ's generation. Similar approach has been applied in most of the work on sequence objectives (Neubig, 2016; Shen et al., 2017; Edunov et al., 2018) to approximate *REINFORCE* algorithm (Williams, 1992).

SeqAlign associates the tokens of model vocabulary voc_θ with their alignment quality $\mathcal{A}(s_\theta^{1..|\theta|}, Y_j)$ and utilizes such quality as the target. Finally, by incorporating the described generation strategy Π^θ, we formulate *SeqAlign* loss as following:

$$\mathcal{L}_{SAlign}(\Theta) = \min\left[\Theta(X_j, \hat{Y}_{j,1..i-1}) - \mathcal{A}(voc_\theta, Y_j)\right]$$
$$\text{where } \hat{Y}_j \sim \Pi^\theta(X_j, \Theta) \quad (5)$$

Note that given the embeddings for all tokens of the vocabulary, this objective can also be formulated as a minimization of the cross-entropy, similarly to *TokenAlign*. We further investigate the impact of the loss formulation in Section 4.3.

3.4 Embeddings Contextualization

Both *TokenAlign* and *SeqAlign* assess the model prediction quality by its alignment to the reference, which require the embeddings of Θ_{emb}. Given the pre-computed embedding vocabulary of *context-insensitive* embedding models, such as GloVe (Pennington et al., 2014) or FastText (Bojanowski et al., 2017), both objectives can be used without further adjustments. However, the use of *context-sensitive* embedding models faces the following issues.

(i) Computation of contextual embeddings re-

quires expensive inference of large language models, such as BERT. Without refinements, an example of obtaining contextual representations for each possible token in generating a 10-token hypothesis, i.e. computing a loss for a single sample would require $10^{|\theta|}$ inferences of Θ_{emb}, where $|\theta|$ is a size of the vocabulary of Θ, commonly in ranges of 30,000–60,000 tokens.

(ii) Bidirectional contextual embeddings inferred in incomplete context are less accurate. Given the exponential growth of hypotheses space, the contextual embeddings can be (a) either inferred within a synthetic context, or (b) inferred incrementally for the each following token using a unidirectional model. We find that both these heuristics significantly alter the pairwise distance of contextual embeddings.

In the *SeqAlign* objective, we address this problem by limiting the embedded vocabulary to the top-n highest-scored tokens of Θ in each prediction step (denoted $\Theta^{\uparrow n}$). By fixing $n = 3$ over our experiments, we need to infer the contextual embeddings of only $\sum_{k=1}^{K} 3|\Pi_k^\theta(X_j)|$ of the highest-scored tokens for each sampled hypothesis $\Pi_k^\theta(X_j)$. In our experiments, we also keep the *number of sampled hypotheses* K fixed to $K = 10$ and we do *not* adjust Θ by the scores of the tokens other than the top ones. As the context, we use the complete hypothesis from which the token $s_\theta^i \in \Theta^{\uparrow n}$ is sampled. Therefore, the alignment $\mathcal{A}$ for distance-based objectives is adjusted as:

$$\mathcal{A}'(s_\theta^i, t_2) = \begin{cases} \mathcal{A}(s_\theta^i, t_2) & \text{if } s_\theta^i \in \Theta^{\uparrow n} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

In *TokenAlign*, which require embeddings of *all* tokens of the vocabulary, we address the computational overhead in a *decontextualization* process. We obtain the decontextualized embedding e^i for each segment s_e^i as an *average* of the contextualized embeddings corresponding to *all* the occurrences of s_e^i in the texts of the training domain X:

$$e_{dec}^i = \Theta_{emb}^{dec}(s_e^i) = \frac{1}{\#s_e^i} \sum_{X_j \in X;\, s_e^i \in X_j} \Theta_{emb}(X_j)^i \quad (7)$$

where $\#s_e^i$ is the number of occurrences of a segment s_e^i in X.

While such process also causes qualitative decay of the contextual representations, it has been shown that decontextualized representations still outperform context-agnostic embeddings in machine translation evaluation (Štefánik et al., 2021). Nevertheless, we further analyze decontextualization impact in Section 4.3.

In our experiments, we use the decontextualized multilingual BERT embeddings (Devlin et al., 2019), extracted from 9-th hidden layer chosen as optimal for evaluation (Zhang et al., 2020b).

4 Methodology

We evaluate the impact of the proposed training objectives in the domain adaptation experiments and compare the results with the adaptation using the commonly-used *MLE* objective as the baseline (§2). We use the novel objectives as the weighted *complements* of the *MLE* objective (Eq. (2)), aiming to extend the modeled space of the problem complexity:

$$\mathcal{L}_{*Align}(\Theta) = \mathcal{L}_{MLE}(\Theta) + \alpha \cdot \mathcal{L}_{NewObj}(\Theta) \quad (8)$$

4.1 Datasets

We choose the data configurations of our experiments to allow the reader to extrapolate trends and conclusions invariant to the covariates of adaptation quality that we consider essential.

Domains. To assess the distributional robustness of the models, we train and evaluate among *all* pairs of the following OPUS domains (Tiedemann, 2012): *Wikimedia*, *OpenSubtitles*, *Bible*, *TEDTalks*, *DGT/Law* and *EMEA/Medical*. We choose the set of domains that reflects both minor (*Wikimedia* → *OpenSubtitles*) and major (*EMEA/Medical* → *Bible*) domain shifts between the training and evaluation. Our selection reflects on real-world settings where practitioners commonly adapt the model to a *specialized* domain such as *law* or *medicine*, but need to keep an operational level of quality on any input.

Data size. We focus on the applications where the size of parallel corpora available for adaptation ranges from *very low-resource* (50,000 aligned sentences, *Bible*) to *medium-resource* (5,100,000 sentences, *DGT/Law*).

Language pairs. Our evaluated language pairs are: *Estonian* → *English*, *German* → *English English* → *Czech*, *English* → *Ukrainian*, *English* → *German* and *English* → *Chinese*. We pick the *English-centric* pairs in order to maximize the number of out-of-domain evaluation sources for the adapted language pair. Our settings cover target languages of Latin, Cyrillic and Chinese alphabets.

4.2 Experimental Setup

Data configuration As the OPUS sources do not contain standard splits, we split the data into train-validation-test. We first de-duplicate the samples and draw 500 validation and 1,000 test samples from each domain.

Hyperparameters & training We perform the adaptations from the bilingual Transformer-base models of Vaswani et al. (2017) using the checkpoints of Tiedemann and Thottingal (2020) pre-trained for a translation of the corresponding language pair on a mixture of OPUS sources.

We perform a hyperparameter search over the parameters of *learning rate*, *objectives weights* α and objective-specific *batch size*. We detail the values and ranges of this search in Appendix B.

After fixing the objectives' parameters, we set up the experiments to closely resemble the traditional training process; We run each experiment until early-stopping by in-domain validation BLEU, with the patience of 20 evaluations, i.e., 10,000 updates and evaluate the model with the best validation score for testing. If the model does not improve over the first 10,000 updates, we evaluate the resulting model after the 10,000 updates.

We implement our experiments using Adaptor library (Štefánik et al., 2022), allowing the release of our implementations in a transparent but self-contained and easy-to-reproduce form.[1]

Evaluation To discourage the effect of the random variance in the performance of the trained model, we report all test scores as the *average* of the performance in the interval of 5 *preceding* and 5 *succeeding* checkpoints, resulting in a single, average test evaluation for each domain.

We collect evaluations of BLEU in the default settings of SacreBLEU (Post, 2018), obtaining a single (average) evaluation of in-domain (ID) BLEU and a set of corresponding evaluations for *all* listed domains *other* than the in-domain (OOD). Given the availability of the sources, this results in four OOD evaluations for all pairs except (en→ukr) and (en→zh) with the datasets for two OOD evaluations.

To enable mutual comparability, we finally normalize both ID and OOD results by the performance of the initial checkpoint and report the change of performance in percentage. We report a single scalar value, or an interval in a form *<mean±range covering all results>*.

[1]Each of our experiments can be reproduced by running a single script; see the README in the attached repository (to be linked here: `github.com/attached/repository`)

4.3 Ablation Experiments

In a set of additional experiments, we estimate the impact of the crucial components of the soft alignment objectives on the adaptation accuracy and robustness. While these assessments are also an ablation study quantifying the impact of our design decisions, significantly, these experiments also assess the impact of different aspects of training of generative language models on their robustness.

Impact of teacher forcing Teacher forcing, i.e. replacing model's own outputs with the preceding tokens of the reference (§2), commonly used in both training and adaptation, circumvents the problem of alignment of the model's generated output to the reference. We suspect that the discrepancy between the training and generation can be magnified under the distribution shift and hence, can be one of the causes of the catastrophic forgetting.

To assess this assumption, we implement a minimal objective conditioning the training by the model's own outputs and compare the difference in the model robustness to *MLE*. We adjust the *SeqAlign* by replacing $\mathcal{A}$ with a *random* alignment as target(s) A_{rand}, while providing the model with its own-generated outputs as prefixes:

$$\mathcal{L}_{SRand}(\Theta) = \min\left[\Theta(X_j, \Pi^{\theta}_{1..i-1}) - \mathcal{A}_{rand}\right] \quad (9)$$

This approach is similar to Choshen et al. (2020), using a *constant* training signal in sequential training and showing the gains similar to expensive MRT maximising BLEU (§2). Additionally, this experiment also quantifies the impact of the embedding-based training signal of *SeqAlign*.

Impact of decontextualization While the *TokenAlign* utilize the *decontextualized* grounding embeddings (§3.4), the decontextualization likely affects the quality of the grounding embeddings, decreasing the quality of such-constructed targets by unknown level.

However, as described in Section 3.4, it is not computationally feasible to simply infer the contextualized embeddings for each candidate token of the generated hypotheses. To allow the comparison of the contextualized and decontextualized version of the same system, we circumvent this problem by adjusting the *SeqAlign*'s alignment $\mathcal{A}'$ (Eq. (6)) to

	Δ BLEU	Bible (de→en) 62,000 pairs	TEDTalks (en→zh) 155,000 pairs	Opensubs (en→ukr) 877,000 pairs	Wiki (en→cze) 1,003,000 pairs	Medical/EMEA (est→en) 1,021,000 pairs	Law/DGT (en→de) 5,105,000 pairs
Orig. BLEU		21.89	29.01	26.12	34.04	54.85	33.56
MLE	ID	**− 8%**	**+ 7%**	+ 4%	+ 9%	+38%	− 1%
	OOD	−53% ± 36%	−23% ± 23%	−15% ± 9%	−15% ± 5%	−35% ± 10%	−19% ± 11%
TokenAlign	ID	−21%	+ 2%	**+ 8%**	**+12%**	**+45%**	+ 1%
	OOD	− 2% ± 1%	**−10% ±12%**	**− 1% ±1%**	**− 6% ±6%**	− 6% ± 7%	**+ 6% ±20%**
SeqAlign	ID	−23%	**+ 7%**	− 8%	+ 8%	+31%	**+ 7%**
	OOD	**− 1% ±1%**	−20% ± 22%	− 2% ± 3%	−12% ± 5%	**− 1% ± 2%**	+ 3% ± 13%

Table 1: **Evaluation of adaptation quality and robustness**: A change of BLEU score relative to the original model, when adapting pre-trained Transformer-base on the titled domain, as measured on a held-out set of the training domain (in-domain, ID) and other listed domains available for the same language pair (out-of-domain, OOD).

utilize the *decontextualized* embeddings instead of the contextualized ones:

$$\mathcal{L}_{\textit{SeqAlign-dec}}(\Theta) = \mathcal{L}_{\textit{SAlign}}(\Theta, \mathcal{A'}_{dec})$$
$$\mathcal{A'}_{dec}(s^i_\theta, t_2) = \min_{e^j_{dec} \in \Theta_{dec}(t_2)} \mathrm{D}(e^i_{dec}, e^j_{dec}) \quad (10)$$

All other parameters (§4.2) remain unchanged.

Impact of the loss formulation While for the sequential objectives, the choice of distance-based loss is compulsed by the lack of alignment $\mathcal{A}$, in our cases, the alignment is known. Hence we can formulate the training objective(s) as the minimization of either a distance loss or a cross-entropy loss.

This analysis evaluates the impact of this choice by introducing an analogous objective to *SeqAlign-dec* (§4.3), which, on the contrary, utilizes the CE loss composing the targets for every predicted token as the quality of its alignment to the reference:

$$\mathcal{L}_{SCE}(\Theta) = \min\left(-\log\frac{\exp(\Theta(X_j, \Pi^\theta_{1..i-1}(X_j)))}{\exp(\mathcal{A}_{dec}(voc_\theta, Y_j))}\right) \quad (11)$$

Identically to *SeqAlign*, we sample the conditioning prefixes from the model's own hypotheses using the stochastic generation strategy Π^θ. To avoid the overhead of inference of contextual embeddings, we also use the alignment $\mathcal{A'}_{dec}$ based on decontextualized embeddings (Eq. (10)).

5 Results

Table 1 compares the results of adaptation using the standard *MLE* objective and our two main objectives: *TokenAlign* and *SeqAlign*, as trained on a selected domain and evaluated on a held-out set of the same domain (ID) and other domains (OOD). The domains are ordered ascending by the size of the training data. Table 2 further aggregates the results per-objective and additionally includes the objectives from our Ablation experiments. More detailed, per-domain results including the ablation objectives can be found in Table 4 in Appendix E.

ΔBLEU:	ID	OOD
0. *MLE*	+ 8% ± 31%	−21% ± 29%
1. *TokenAlign*	**+ 9%±30%**	− 2% ± 9%
2. *SeqAlign*	+ 3% ± 27%	**− 1%± 8%**
3. *SRand*	+ 3% ± 31%	− 6% ± 5%
4. *SeqAlign-dec*	+ 5% ± 31%	− 6% ± 27%
5. *SCE*	+ 4% ± 32%	−17% ± 44%

Table 2: **Results of Ablation experiments**: Average change of BLEU scores relative to the original model, when adapting Transformer-base model with a given objective. The intervals cover the averages of 6 in-domain and 20 out-of-domain evaluations (§4.2).

Alignment-based objectives improve robustness Both *TokenAlign* and *SeqAlign* objectives consistently improve the model robustness (OOD) over the *MLE* in *all* the evaluated cases. In addition, comparing *TokenAlign* to *MLE*, we also see the advances in the adaptation quality (ID), in three out of four cases where *MLE* was able to deliver any ID improvements. In ID performance, *SeqAlign* is the only one able to utilize the higher resource availability of the *Law/DGT* domain, but lacks in ID substantially on *Medical/EMEA* domain. In OOD evaluations, *SeqAlign* performs comparably to *TokenAlign*. Nevertheless, all objectives remain to fail to adapt in very low-resource adaptation of a significant domain shift (*Bible*).

While the results confirm our main hypothesis that circumventing *MLE*'s assumption of a single-truth prediction largely improve model's distribu-

tional robustness, we observe discrepancies in in-domain performance over different sizes of the training data similar to *MLE*. Even though *SeqAlign* utilizes larger volumes of conditioning prefixes, its performance on the two smallest domains is inferior to both *TokenAlign* and *MLE*, while on the contrary, it is the most efficient among objectives in medium-resource *Law/DGT*. This could be a consequence of the lower quality of the model's self-generated prefixes under large domain shifts (*Bible* domain).

Avoiding teacher-forcing improves robustness A comparison of the results of *SRand* and *MLE* in Table 2 shows that the mere exposition of the model to its own hypotheses reduces the forgetting of *MLE* by 71% in average ($-21\% \rightarrow -6\%$). However, constructing the non-informative targets for self-generated inputs also causes a decay in adaptation quality ($+8\% \rightarrow +3\%$).

Alignment-based targets complement avoiding teacher-forcing A comparison of the results of *SRand* to *SeqAlign* (Table 4 in Appendix E) shows robustness superiority of *SeqAlign* in four out of five scenarios, suggesting that the enhancements in robustness might be attributed both to the semantically-constructed targets and avoidance of the teacher forcing. While the aggregate in-domain results of *SeqAlign* and *SRand* in Table 2 are very similar, the per-domain results reveal that their results vary over domains and the suggested ID tie of *SRand* to *SeqAlign* and is largely attributed to *SRand*'s better result on *Bible*, where both objectives fail to improve ID nevertheless.

Decontextualization does not carry a large qualitative drop Both objectives grounding its targets in decontextualized embeddings (*TokenAlign* and *SeqAlign-dec*) show relatively good average results on both in-domain and out-of-domain (Table 2), where *TokenAlign* is the only objective reaching in-domain gains *superior* to *MLE* in average. A comparison of *SeqAlign* to its decontextualized instance (*SeqAlign-dec*) specifically evaluates the impact of decontextualization, in the settings of absolute distance loss and no teacher forcing. We see that while the decontextualization leads to a relatively large average loss in the robustness ($-1\% \rightarrow -6\%$), *SeqAlign-dec* outperforms *SeqAlign* on the in-domain ($+3\% \rightarrow +5\%$). Per-domain results (Table 4 in Appendix E) show that this is attributed mainly to the superior adaptation performance of *SeqAlign-dec* in the low-resource *Opensubs (en→ukr)* domain, suggesting that the averaging of decontextualization might also have a *denoising* effect in the low-resource settings. This case opposes our suspicion that decontextualization by embeddings' averaging might produce quality representations only in higher-resource settings.

Loss formulation impacts model robustness A comparison of *SeqAlign-dec* and *SCE* in Table 2 assesses the difference in performance when varying the loss formulation in the sequence alignment objective. The difference is significant in OOD evaluation, where changing a distance-based loss to the entropy-based causes a significant drop ($-6\% \rightarrow -17\%$), comparable to the drop of the traditional *MLE*, also built upon CE loss (-21%). However, the superior performance of CE-based *TokenAlign* contradicts that distance-based loss is always a better choice and optimal selection of the loss remains convoluted by other covariates.

6 Conclusion

Our work sets out to explore the alternatives between the efficient yet naive *MLE* objective and expressive but resource- and computationally-demanding sequential objectives, building the training signal in the alignment of the semantic token representations. We build an alignment mechanism applicable with any chosen embedding system and propose two main objectives that utilize the constructed alignment as its target; either (i) keeping or (ii) circumventing the teacher-forcing of the reference in training. We find that both approaches persist robustness of the adapted model much better than the traditional approach while obtaining comparable results in the quality of adaptation.

We thoroughly investigate the impact of selected design choices on the robustness of generative LLMs in the ablation experiments. Among others, we find that a relatively large portion of the model's robustness can be recovered by including the model's own outputs among the inputs. Future work might also benefit from the qualitative assessment of the impact of the decontextualization eliminating the computational overhead of applying the contextualized embeddings in dynamic contexts.

We look forward for future work that will explore the potential of applying semantically-grounded objectives in a more robust and efficient pre-training and adaptation for numerous other applications of language models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, USA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the ACL*, 5:135–146.

Fredrik Carlsson, Joey Öhman, Fangyu Liu, Severine Verlinden, Joakim Nivre, and Magnus Sahlgren. 2022. Fine-grained controllable text generation using non-residual prompting. In *Proceedings of the 60th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 6837–6857, Dublin, Ireland. ACL.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving Sequence-to-Sequence Learning via Optimal Transport. *ArXiv*, abs/1901.06283.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. ACL.

Praveen Dakwale and Christof Monz. 2017. Fine-Tuning for Neural Machine Translation with Limited Degradation across In- and Out-of-Domain Data. In *Proceedings of the XVI Machine Translation Summit (Vol. 1: Research Track)*, pages 156–169, Nagoya, Japan.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the NAACL: Human Language Technologies*, pages 4171–4186, Minneapolis, USA. ACL.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical Structured Prediction Losses for Sequence to Sequence Learning. In *Proceedings of the 2018 Conference of the NAACL: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. ACL.

Thierry Etchegoyhen, Anna Fernández Torné, Andoni Azpeitia, Eva Martínez Garcia, and Anna Matamala. 2018. Evaluating Domain Adaptation for Machine Translation Across Scenarios. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. ELRA.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *ArXiv*.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. *CoRR*, abs/1312.6211.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From Word Embeddings To Document Distances. In *Proc. of International Conference on Machine Learning*, volume 37, pages 957–966, Lille, France. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. of the 58th Annual Meeting of the ACL*, pages 7871–7880.

Wenjie Lu, Leiying Zhou, Gongshen Liu, and Quanhai Zhang. 2020. A mixed learning objective for neural machine translation. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 974–983, Haikou, China. Chinese Information Processing Society of China.

Chenggang Mi, Lei Xie, and Yanning Zhang. 2020. Improving Adversarial Neural Machine Translation for Morphologically Rich Language. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):417–426.

Graham Neubig. 2016. Lexicons and Minimum Risk Training for Neural Machine Translation: NAIST-CMU at WAT 2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 119–125, Osaka, Japan. The COLING 2016 Organizing Committee.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. 2015. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning. ArXiv:1412.6614.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the EMNLP*, pages 1532–1543, Doha, Qatar. ACL.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Lariviere, Alina Beygelzimer, Florence d'Alche Buc, Emily Fox, and Hugo Larochelle. 2021. Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program). *Journal of Machine Learning Research*, 22(164):1–20.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. ACL.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence Level Training with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings*.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary Adaptation for Domain Adaptation in Neural Machine Translation. In *Findings of the ACL: EMNLP 2020*, pages 4269–4279. ACL.

Danielle Saunders. 2021. Domain Adaptation and Multi-Domain Adaptation for Neural Machine Translation: A Survey. *CoRR*, abs/2104.06951.

Christophe Servan, Josep Maria Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for Neural Machine Translation. *ArXiv*, abs/1612.06141.

Shiqi Shen, Yong Cheng, Zhongjun He, W. He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th ACL (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. ACL.

Shiqi Shen, Yang Liu, and Maosong Sun. 2017. Optimizing Non-Decomposable Evaluation Metrics for Neural Machine Translation. *Journal of Computer Science and Technology*, 32:796–804.

Michal Štefánik, Vít Novotný, Nikola Groverová, and Petr Sojka. 2022. Adaptor: Objective-Centric Adaptation Framework for Language Models. In *Proceedings of the 60th Annual Meeting of the ACL: System Demonstrations*, pages 261–269, Dublin, Ireland. ACL.

Michal Štefánik, Vít Novotný, and Petr Sojka. 2021. Regressive ensemble for machine translation quality evaluation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1041–1048. ACL.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming Catastrophic Forgetting During Domain Adaptation of Neural Machine Translation. In *Proceedings of the 2019 Conference of the NAACL: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. ACL.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proc. of the Eighth International Conf. LREC*, pages 2214–2218, Istanbul, Turkey. ELRA.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. EAMT.

Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. 2021. BERTTune: Fine-Tuning Neural Machine Translation with BERTScore. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNL (Volume 2: Short Papers*, pages 915–924. ACL.

Michael Ustaszewski. 2019. Exploring Adequacy Errors in Neural Machine Translation with the Help of Cross-Language Aligned Word Embeddings. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)*, pages 122–128, Varna, Bulgaria. Incoma Ltd., Shoumen, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proc. of the 31st NIPS conference*, volume 30 of *NIPS '17*, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the ACL*, pages 3544–3552. ACL.

Yiyu Wang, Jungang Xu, and Yingfei Sun. 2022. End-to-End Transformer Based Model for Image Captioning.

Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proc. of the 2020 Conf. EMNLP: System Demonstrations*, pages 38–45. ACL.

Weijia Xu, Xing Niu, and Marine Carpuat. 2019. Differentiable Sampling with Flexible Reference Word Order for Neural Machine Translation. In *Proceedings of the 2019 Conference of the NAACL: Human*

Language Technologies, Volume 1 (Long and Short Papers), pages 2047–2053, Minneapolis, Minnesota. ACL.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. In *Proceedings of the 2018 Conference of the NAACL: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana. ACL.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *CoRR*, abs/1609.05473.

Ruiyi Zhang, Changyou Chen, Xinyuan Zhang, Ke Bai, and Lawrence Carin. 2020a. Semantic Matching for Sequence-to-Sequence Learning. In *Findings of the ACL: EMNLP 2020*, pages 212–222. ACL.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating Text Generation with BERT. In *Proc. of International Conference on Learning Representations*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 4334–4343, Florence, Italy. ACL.

A Limitations

Our work experiments with a range of adaptation domains that we draw systematically to capture the covariates enumerated in Section 4.1. However, future work should acknowledge that these are not all the covariates responsible for the success of adaptation and the robustness of the final model. Following is the non-exhaustive list of possible covariates that we do not control in this work. (i) the adapted model size, (ii) the size of pre-training data, (iii) pre-training configuration parameters, but also (iv) the broad variance of adapted language pair(s); (v) the variance of mutual similarity of languages within the pair, and hence (vi) the difficulty of training the translation model.

To avoid difficulty with normalizing BLEU values over different writing systems, we did not perform our experiments on languages using other than Latin and Cyrillic script and hence, our results are not representative of some major languages such as Chinese or Arabic. However, the alignment approach presented in Section 3.1 and adapted by all the proposed objectives is also applicable to other writing systems.

The evaluation of our experiments did not consider the effect of *randomness* of the training process. Despite the fact that our experiments were run with a fixed random seed and initial value, making our results deterministically reproducible, the variance of the results among the experiments of different random seeds was not investigated due to the related infrastructural costs. However, all our results are aggregated over larger set of checkpoints and/or domains, ranging from 10 (IDs in Table 1) to 720 (OODs in Table 2), as described in Section 4.2.

The alignment scheme proposed in Section 3.1 has known biases; for instance, in the cases utilizing decontextualized embeddings, where both the hypothesis and reference contain the multiple occurrences of the same word, the alignment scheme will make the prediction of the same target token equally *good*, regardless of the position. This flaw could be further addressed by using the Optimal transport alignment (Kusner et al., 2015), similarly to Zhang et al. (2020a).

B Hyperparameter search

For each of the evaluated objectives, we perform a hyperparameter search independently over the selected parameters in the denoted range, based on the best in-domain validation BLEU reached in the adaptation to *Wikimedia* domain.

(1) **learning rate**: ranging from $2 \cdot 10^{-7}$ to $2 \cdot 10^{-4}$, with step 10. (2) **objectives ratio** α (Eq. (8)): we manually set the weight of the additional objective such that the loss values for both components of the final loss are approximately balanced, based the first 10 valuations. We do not perform further tuning and use the same weights over all experiments. (3) **Batch size**: For *ML* experiments, we fix the effective batch size to 60, we pick the optimal batch size for *TokenAlign* and *SeqAlign* objectives over $[1, 5, 10, 20]$.

Other parameters that we adjust and remain fixed over the experiments are following: $\textbf{warmup steps} = 1,000$, **LR schedule** as *constant decay*. Distance-based objectives including *SeqAlign* introduce two new parameters: (i) K: a number of the sampled hypotheses and (ii) n: a number of most-likely tokens to align. To keep the computation time feasible, we do not perform further tuning and set these parameters to $K = 10$ and $n = 3$ over all the experiments. All other parameters can be retrieved from the defaults of

TrainingArguments of Transformers (Wolf et al., 2020), version 4.10.2.

We treat the optimized hyperparameters as *independent*; hence we optimize each variable separately. Our configuration results in experimenting with 9 hyperparameter search runs for each objective, including *MLE* baseline.

C Computational demands

We performed the adaptation of each of the proposed objectives on a server with a single *NVidia Tesla A100*, 80 GB of graphic memory, 512 GB of RAM and 64-Core Processor (*AMD EPYC 7702P*). We also tested to train all our experiments using lower configuration using a single *NVidia Tesla T4*, 16 GB of graphic memory, 20 GB of RAM and a single core of *Intel(R) Xeon(R)* processor.

We benchmark the running times of the time-demanding parts of the adaptation process in the first-mentioned configuration. We find that the proposed decontextualization process required by *TokenAlign*, *SCE* and *SeqAlign-dec* takes in these settings between 50 minutes on the smallest domain to 25 hours on the largest domain. Table 3 shows the average speed of updates and a number of steps that each of the designed objectives requires to converge. Further details on our methodology are described in Section 4.2.

Objective	Updates / hour	Updates to converge
MLE	451	15,500
TokenAlign	404	24,000
SeqAlign	287	11,875
SRand	152	10,100
SeqAlign-dec	295	7,500
SCE	585	23,740

Table 3: **Adaptation speed**: Average number of updates per hour and average number of updates to converge that we measure over objectives in our experiments.

D Details of Alignment Algorithm

Algorithm 1 describes the alignment procedure that we propose to obtain *grounding embeddings* for the tokens of the trained model.

Our approach first *aligns* the model and embeddings vocabulary; Given a text t, we obtain two ordered sequences of textual segments (tokens): grounding embeddings tokens $s_e(t)$ and model tokens $s_\theta(t)$. We obtain the *model grounding embeddings* e^i_θ of each *model* segment $s^i_\theta \in s_\theta(t)$ to each *grounding* segment $s_{e,i} \in s_\theta(t)$ by (i) assigning the *coverage intervals* of t to each model and embedding segment $s_\theta(t)$ and $s_e(t)$, and (ii) for each model segment $s^i_\theta \in s_\theta(t)$, searching for the segment $s^i_e(t)$ with *largest intersection* of the covering intervals $|s^i_\theta \cap s^j_e|$.

proc *align_to_grounding*(s_θ, s_e):
  foreach $i \in 1..|s_\theta|$ **do**
    while $|s^i_\theta \cap s^j_e| > best\_cov$ **do**
      $pairs_i \leftarrow j$
      $best\_cov \leftarrow |s^i_\theta \cap s^j_e|$
      $j \leftarrow j + 1$
  return $pairs$

Algorithm 1: Ability to pair each model token s^i_θ with the best-matching grounding segment s^j_e allows us to use alignment grounded in domain-agnostic representations. Relying on the consistent ranking of the aligned sequences, the grounding alignment algorithm requires at most $(|s_\theta| + |s_e|)$ steps to finish.

E Detailed results of all objectives

Table 4 shows a comparison of *all* objectives over all evaluated domains, providing a finer-grained report of results presented in Table 2. Note that in order to eliminate the effect of different scaling of BLEU evaluations in character-segmented BLEU results, we exclude the (en→zh) pair from the ablations. The methodology of results collections is described in Section 4.2. The discussion including these results is present in Section 5.

F Training validation reports

We report and compare the change of validation BLEU of our two main objectives, relative to the *MLE* objective over the course of our experiments and overview the results in Figures 5 and 6 for *SeqAlign* and *TokenAlign* objective, respectively.

The plots aggregate 5 training logs and their corresponding out-of-domain logs into the in-domain and out-of-domain reports, for easy comparability with *MLE*, both in-domain and out-of-domain BLEUs of *MLE* are *averaged* and paired with the corresponding BLEUs of the inspected objective over the shared evaluation domain. Finally, the plots of the inspected objective consist of *50% quantile intervals* and the *average* of BLEU rel-

	Δ BLEU	Bible (de→en) 50,000 pairs	Opensubs (en→ukr) 80,000 pairs	Wiki (en→cze) 100,000 pairs	Medical/EMEA (est→en) 300,000 pairs	Law/DGT (en→de) 5,100,000 pairs
Orig. BLEU		21.89	26.12	34.04	54.85	33.56
MLE	ID	− 8%	+ 4%	+9%	+38%	− 1%
	OOD	−53% ± 36%	−15% ± 9%	−15% ± 5%	−35% ± 10%	−19% ± 11%
TokenAlign	ID	−21%	+ 8%	**+12%**	**+45%**	+ 1%
	OOD	− 2% ± 1%	**− 1% ± 1%**	**− 6% ± 6%**	− 6% ± 7%	**+ 6% ± 20%**
SeqAlign	ID	−23%	− 8%	+ 8%	+31%	**+ 7%**
	OOD	**− 1% ± 1%**	− 2% ± 3%	−12% ± 5%	**− 1% ± 2%**	+ 3% ± 13%
SRand	ID	−14%	− 7%	+ 8%	+34%	− 7%
	OOD	− 8% ± 2%	− 3% ± 3%	− 9% ± 3%	− 7% ± 5%	− 7% ± 5%
SeqAlign-dec	ID	−26%	**+11%**	+ 5%	+35%	+ 2%
	OOD	−13% ± 8%	− 1% ± 1%	−11% ± 19%	−12% ± 7%	+ 4% ± 17%
SCE	ID	**+ 8%**	+ 9%	+11%	+ 1%	−11%
	OOD	−78% ± 9%	−32% ± 1%	−12% ± 5%	− 1% ± 2%	−14% ± 13%

Table 4: **Evaluation of adaptation quality and robustness over *all* designed objectives**: A change of BLEU score relative to the original model, when adapting pre-trained Transformer-base on selected domain, as measured on a test set of the training domain (in-domain, ID) and out-of-domain (OOD). The aggregates over all domains are listed in Table 2.

ative to both the *MLE* BLEU and initial model performance. Note that while the relative distances of *MLE* to the corresponding plots of the other objective *always* correspond, some training runs are terminated in the course of the plotted steps, explaining some sudden performance gains in the plot.

While the performance decay of *MLE* by the time of early-stopping by in-domain BLEU is close-to linear, *TokenAlign* in average maintains none, or minimal decays of the out-of-domain performance, although the variance of the initial decay significantly varies over domains. This trend implies that the early-stopping strategy based on in-domain performance does not significantly decay the robustness results and favours the deployment of *TokenAlign* in situations where no validation out-of-domain data is present.

The robustness of the model trained using *SeqAlign* behaves differently and the initial robustness decay is more significant. However, the decay soon diverges from *MLE* and noticeably, after the 5,000-th step *all* the robustness evaluations of *SeqAlign* report robustness gains over *MLE*.

Although we restrain from drawing conclusions based exclusively on these plots, the comparisons suggest that while the decay of robustness of *MLE* training is continuous, in the case of soft objectives, the decay gradually slows, while the model incrementally reaches potential in-domain gains similar to *MLE*.

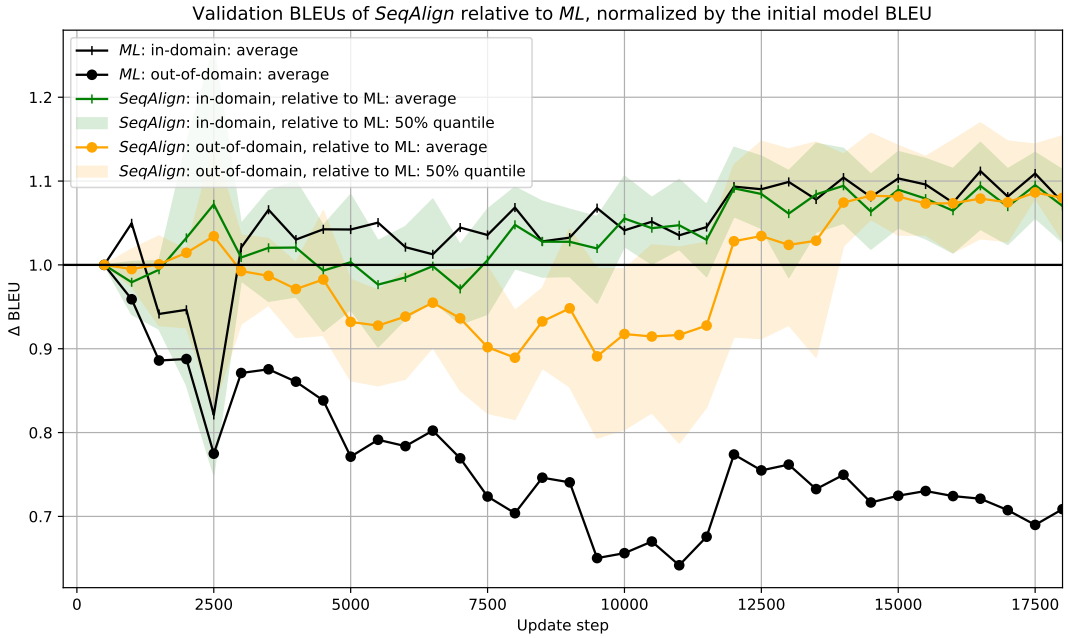

Figure 5: Comparison of **validation BLEU of *MLE* and *SeqAlign* objective** reported over the training on 5 different domains and 20 corresponding out-of-distribution domains until the in-domain early-stopping. For easier comparison, both *MLE* logs are averaged and reported intervals correspond to the 50%-quantile of difference to the *MLE* run on the corresponding evaluation domain. While the training with *MLE* objective consistently magnifies the *forgetting* of adaptation, the soft objectives report a higher OOD score over all experiments while reaching comparable adaptation gains on the in-domain. Note that the two major gains of *SeqAlign* before steps 12,000 and 14,000 are attribute to early-stopping of specific runs at these points and hence, should be excluded from the conclusions. See Appendix F for further description.

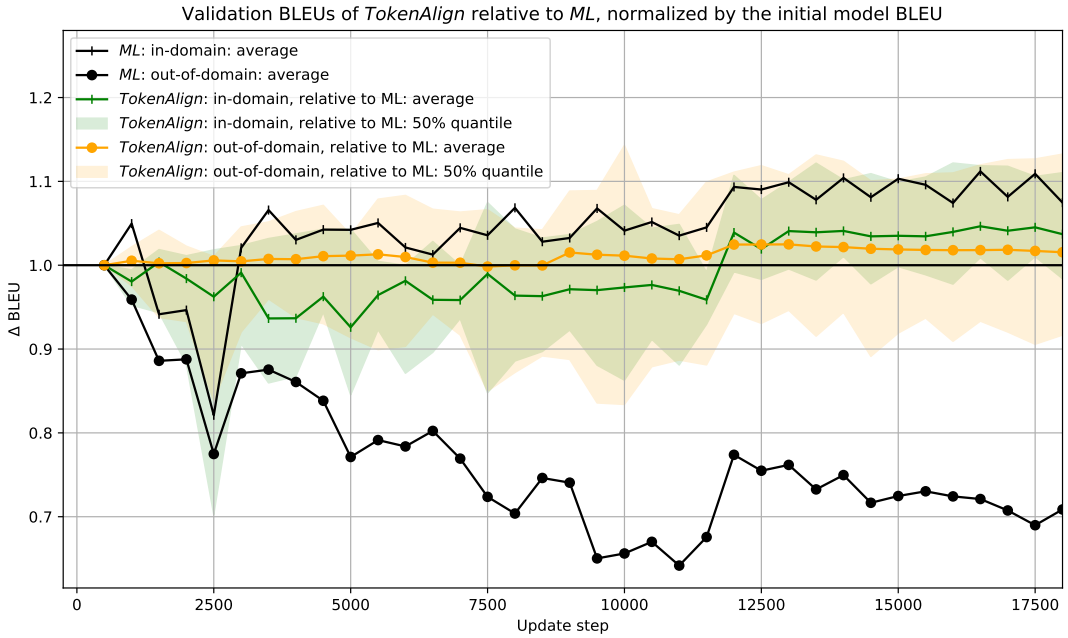

Figure 6: Comparison of **validation BLEU of *MLE* and *TokenAlign* objective** as reported over the training on 5 different domains and 20 corresponding out-of-distribution domains until in-domain early-stopping. See Figure 5 and Appendix F for further description.