Bridging Domains for Automatic Post-Editing: A Classifier-Guided Multi-Domain Adaptation Framework

Sourabh Deoghare, **Diptesh Kanojia** and **Pushpak Bhattacharyya**
CFILT, Indian Institute of Technology Bombay, Mumbai, India
Institute for People-Centred AI, University of Surrey, United Kingdom
{sourabhdeoghare, pb}@cse.iitb.ac.in, d.kanojia@surrey.ac.uk

Abstract

Automatic Post-Editing (APE) is a widely studied approach for enhancing the output quality of Neural Machine Translation (NMT) systems. While most prior work has focused on general-purpose APE, the potential of domain-specific APE, such as for personalized or specialized content, remains underexplored due to the scarcity of domain-labeled training data. In this work, we investigate domain adaptation for APE using adapter-based methods. Our proposed multitask learning-based domain adaptation framework includes the use of a domain classifier to get a weighted combination of parallel domain-specific adapters at inference time, without requiring prior domain knowledge. This design allows the model to leverage cross-domain similarities, making it especially robust in low-resource domain scenarios. Our experimental results on English–German, English–Marathi, and English–Tamil pairs across different domains for each pair show substantial improvements over their respective general-purpose APE baselines. To facilitate further research, we will release human-annotated domain labels for triplets in WMT22 English–Marathi, and WMT24 English–Tamil APE datasets and the code.

1 Introduction

Automatic Post-Editing (APE) focuses on developing computational approaches to improve Machine Translation (MT) system-generated output by following the principle of minimal editing (Bojar et al., 2015; Chatterjee et al., 2018). While APE systems have shown success in enhancing translation quality, most existing research focuses on developing general-purpose models trained on large-scale, domain-agnostic data (Bhattacharyya et al., 2023; Zerva et al., 2024).

While effective in broad settings, such models fail to address domain-specific nuances like unique terminologies, and sentence style, critical for applications such as tourism, medical, legal, or personalized translation. Adapting APE systems to specific domains is thus a valuable but unexplored direction, largely due to the scarcity of adequately sized domain-labeled post-editing data and also the difficulty of maintaining separate models for each domain.

A promising approach to addressing the challenges of domain variability in Automatic Post-Editing (APE) is adapter-based learning, which facilitates efficient fine-tuning of a pretrained model for multiple domains without requiring full retraining (Huang et al., 2022). To the best of our knowledge, this is the first work that systematically evaluates domain-adapted APE systems in a domain-specific manner, highlighting the strengths and weaknesses of adaptation strategies across varied domains. While prior work has explored classifier-guided adapter-based architectures, these typically rely on a classifier to select a single adapter at inference time. In contrast, our method leverages the classifier's output to compute a weighted combination of all domain adapters, enabling the model to aggregate cross-domain signals rather than restricting it to a single domain representation. This design is particularly beneficial for real-world, mixed-domain scenarios, where explicit domain labels

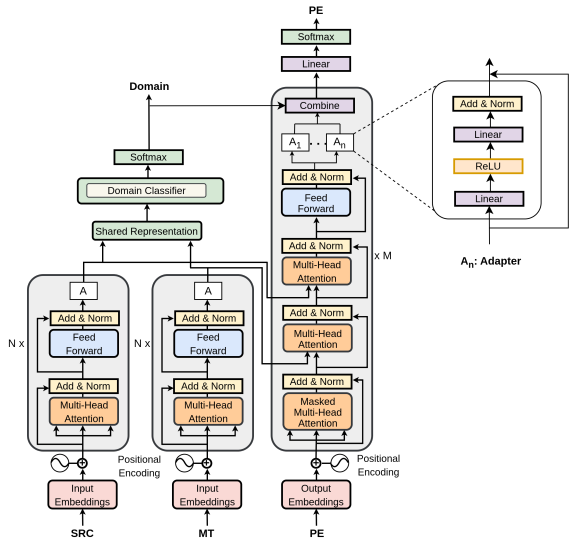

Figure 1: APE model architecture of AdaptAPE, the proposed domain adaptation framework. Blocks A, A_1 to A_n denote adapter modules, where n is number of domains.

may be unavailable or ambiguous. Furthermore, our evaluation framework is carefully constructed to assess the generalization ability of the proposed approach across diverse language pairs and domain settings. Specifically, we conduct experiments on English–German, English–Marathi, and English–Tamil, which differ substantially in terms of linguistic typology (Germanic, Indo-Aryan, Dravidian), domain availability, data distribution, and the degree of domain divergence, allowing us to demonstrate robust generalization across languages and domains.

Our contributions are:

- A novel classifier-guided, multi-adapter-based APE domain adaptation framework, where the classifier is used not to select a single adapter at inference time, but to compute a weighted combination of all domain adapters, enabling more effective post-editing through cross-domain knowledge integration. Our approach outperforms the respective general-purpose baselines by $0.93 - 1.12$, $0.48 - 2.40$, and $3.55 - 12.19$ TER points across various domains for En–De, En–Mr, and En–Ta pairs (Refer Tables 3, 4, 5).

- Human-annotated domain labels for En–De, En–Mr, and En–Ta WMT APE datasets. Each APE triplet is annotated with a single domain label. We will release these annotations to encourage further APE domain adaptation research (Refer Section 3).

- A comparative analysis of adapter-based domain adaptation techniques in the context of APE, giving insights into the injection of adapters on the encoder and decoder sides and cross-domain similarity exploitation (Refer Section 5).

2 Related Work

We see initial attempts in dealing with domain mismatch when the WMT20 APE shared task changed the domain for English–German from IT to Wikipedia (Chatterjee et al., 2020).

Lee at al. (2020) used transfer learning by including the IT domain data during the pre-training stage and then fine-tuning the model on the Wikipedia domain data. The other submissions to the WMT20 APE shared task took the route of artificial APE data generation rather than exploring domain-adaptation or transfer learning-based solutions (Chatterjee et al., 2020). Similarly, Sharma et al. (2021) used extracted in-domain data during the initial stages of training and used the out-of-domain human-annotated data during the fine-tuning stage due to its high quality. However, the study has not reported performance on the out-of-domain data. Deoghare et al. (2024) showed that jointly fine-tuning a multilingual APE model on APE and Quality Estimation (QE) tasks using in-domain data leads to APE performance improvements.

The submission by Huang et al. (2022) to the WMT22 English–Marathi APE shared task (Bhattacharyya et al., 2022) focused on multi-adapter- and domain classifier-based model architecture. Multiple parallel adapters, one for each domain, are added to a transformer decoder. Cross-lingual representation generated by the transformer encoder is additionally passed to a domain classifier, which is later used to activate the corresponding domain-specific adapter in each decoder layer during inference. The model is trained in a sequential manner: initially, a standard transformer is trained for APE; in the second step, adapter modules are injected into the decoder; the third step involves learning parameters of the domain classifier; and finally, only the adapter module weights are updated, with the rest of the parameters frozen. The domain adaptation framework proposed in our work closely resembles this architecture.

Notably, none of these existing APE domain-adaptation approaches evaluate performance on domain-specific evaluation sets, making it difficult

to assess their true effectiveness in handling domain variability.

3 AdaptAPE: APE Domain Adaptation Framework

The section describes the model architecture, data, and training procedure used in the proposed framework.

3.1 Architecture

Figure 1 illustrates the APE model architecture employed in our framework. We adopt a two-encoder, single-decoder Transformer architecture, where the source sentence and the MT-generated translation are processed by separate encoders, following Deoghare et al. (2024). The combined representation from the two encoders is passed to the decoder for post-edit generation and simultaneously fed to a classifier for domain classification.

A single adapter module (domain-agnostic) (Bapna and Firat, 2019) is inserted into each encoder block. This design choice enables targeted learning during domain classifier training: all encoder layers are frozen, and only the adapter parameters are updated. As a result, these adapter modules learn to extract features that are useful for domain classification without altering the underlying encoder representations. Similarly, parallel domain-specific adapter modules are injected in parallel into each decoder block. The domain classifier, placed on top of the encoder stack, processes the combined encoder representation through a feed-forward layer of size 512 with a *tanh* activation, followed by a final feed-forward layer whose output dimension equals the number of domains. A *softmax* activation is then applied to produce a probability distribution over the domains.

Each adapter module consists of a down-projection layer, a ReLU activation, and an up-projection layer. The adapter output is added to the block input via a residual connection, followed by a layer normalization step. A skip connection is also incorporated, allowing the model to bypass the adapter entirely when necessary, thereby supporting stable training and improved generalization.

The outputs from the multiple domain-specific adapters in each decoder block are first combined (e.g., summed or concatenated and projected) to match the input dimensionality before being passed to the next block. In the final decoder block, the outputs of all domain adapters are combined using weights derived from the domain classifier's output. This weighted representation is then passed through a linear transformation followed by a softmax layer to generate the vocabulary distribution for post-edit generation.

Pair	Split	General	Health	Legal	Tourism	Wiki	IT
En–De	**Train**	-	-	-	-	7000	13442
	Test	-	-	-	-	1000	1000
En–Mr	**Train**	7000	5000	-	6000	-	-
	Dev	368	317	-	315	-	-
En–Ta	**Train**	3073	2295	1632	-	-	-
	Dev	500	250	250	-	-	-

Table 1: Statistics of the real (human-annotated) APE data. Wiki: Wikipedia

3.2 Training Procedure

We follow the curriculum training strategy as used in earlier works (Oh et al., 2021; Deoghare et al., 2024) to train an initial APE model and use multitask learning during the final stage, where the model is jointly trained on APE and classification tasks with updates made only to the classifier and adapter layers.

The first step involves training a single encoder, single decoder model for the NMT task using the parallel corpus. In the second step, an encoder for encoding MT-generated translation is added, and resulting two-encoder, single decoder model is trained for APE using all out-of-domain synthetic APE triplets and in-domain synthetic APE triplets with TER greater than the average TER score of the human-generated APE training set. In the next step, we inject the domain classifier on top of the encoder and also inject the single-adapter layers into the encoder and the multi-adapter layers into the decoder. The resulting model is then trained to learn the parameters of only these added modules for the APE task using in-domain synthetic APE data with a TER less than the average TER of the human-generated training set. We use Nash-MTL (Deoghare et al., 2023) to perform joint training since simply adding losses of both tasks may lead to overfitting on the domain classification task soon before the model converges for APE.

3.3 Data

The framework uses publicly available parallel corpora, WMT-released synthetic and real APE datasets. In addition, the framework requires a domain label for each APE triplet. For the English–German (En–De) pair, WMT has released separate single-domain datasets (Chatterjee et al., 2019;

Pair	General	Health	Legal	Tourism
En–Mr	8302	4645	-	5053
En–Ta	3673	2081	1246	-

Table 2: Statistics of the En–Mr and En–Ta Train splits single domain annotated through LLaMA-3.3-70B-Instruct.

Chatterjee et al., 2020). However, for English–Marathi (En–Mr) and English–Tamil (En–Ta) pairs, while the synthetic APE data is domain-separated, WMT has only mentioned the domains covered in the overall real APE datasets (Bhattacharyya et al., 2022; Zerva et al., 2024).

Therefore, for these two pairs, we obtained a single domain label for each real APE triplet in the training and development sets through a single professional translator for each pair. The annotators were provided with the list of possible domains for each language pair and instructed to assign the most appropriate label per instance.

Table 1 shows the size of per-domain data in train or evaluation splits for each of these pairs. Since WMT has not released the test sets for En–Mr and En–Ta pairs, we use the respective development (Dev) sets for the evaluation. For the En–De pair, the corresponding evaluation sets are used for the evaluation. The WMT22 English–Marathi training set consists of 7,000, 6,000, and 5,000 instances from the General, Tourism, and Health domains, respectively. The corresponding development set includes 368 General, 315 Tourism, and 317 Health domain instances. Similarly, the WMT24 English–Tamil training set contains 3,073 General, 2,295 Health, and 1,632 Legal domain triplets. The associated development set comprises 500 General, 250 Health, and 250 Legal domain triplets.

Details of all datasets used during the training are discussed in **Appendix B**.

We also employ LLaMA-3.3-70B-Instruct[1] and use 5-shot prompting to annotate the En–Mr and En–Ta APE triplets in the training sets, with the goal of evaluating whether human-annotated domain labels offer additional value. The prompt template used for domain classification is provided in **Appendix J**, and the resulting domain-wise data distribution is summarized in Table 2.

We observe that LLaMA-3.3-70B-Instruct often misclassifies a substantial number of Tourism and Legal domain sentences as belonging to the General domain. This behavior is typical of Large Language Models (LLMs) when domain-specific cues are subtle, implicit, or absent. Many such sentences use neutral vocabulary that overlaps with multiple domains and lack explicit indicators such as named entities, technical terminology, or culturally grounded references. Moreover, in the absence of a broader discourse context, the model may default to the General domain due to classifier uncertainty. In contrast, human annotators relied on their world knowledge, contextual reasoning, background knowledge, and cultural associations, like recognizing descriptions typical of tourist artifacts or legal processes, to more accurately assign the appropriate domain, even when explicit cues are limited. **Appendix H** shows such misclassified examples.

Additionally, to test the out-of-domain performance of the domain-adapted APE models, we set aside 500 Administrative/Governance domain sentence pairs from the En–Mr synthetic APE data. These sentences were shortlisted where the cosine similarity between the source and reference/post-edit representations obtained through LaBSE (Feng et al., 2022) is more than 0.85. These triplets were not used during any stage of the training.

3.4 Difference between AdaptAPE and Huang et al. (2022)

The proposed framework differs from (Huang et al., 2022) in the following ways. First, while their approach adds adapters only to the decoder, we incorporate adapters into both the encoder and decoder. This allows domain information to influence the representation of source sentences and enhances the domain classifier's accuracy by providing more discriminative, domain-aware input features. Second, instead of training the domain classifier and adapters in separate stages, our model is trained jointly using a multitask learning objective that optimizes both APE and domain classification tasks. This encourages better coordination between domain prediction and post-editing. Finally, rather than relying on a hard selection of a single domain adapter, we use the domain classifier to compute a probability distribution over domains, which is then used to weigh the outputs of multiple domain-specific adapters in the final decoder layer. This enables a more flexible and expressive adaptation mechanism, particularly beneficial for handling ambiguous or mixed-domain inputs.

[1]LLaMA-3.3-70B-Instruct

Experiment	Wikipedia		IT	
	TER	BLEU	TER	BLEU
Do Nothing	18.05	71.07	15.08	76.94
w/o Adaptation	17.87	71.24	15.74	76.31
Transfer Learning	17.80	71.38	15.98	76.05
Single Adapter (Dec)	17.42	71.76	15.65	76.40
Single Adapter (Enc + Dec)	17.16	72.00	15.49	76.55
AdaptAPE w/o Enc	17.17	72.01	14.89	77.07
AdaptAPE	**16.94**	**72.14**	**14.62**	**77.42**
Huang et al. (2022) (Joint)	17.25	71.87	15.06	76.88
LLM-APE w/o Adaptation	18.51	70.29	15.30	76.42
LLM-APE w/ Adaptation (Single)	17.95	71.16	15.29	76.46
LLM-APE w/ Adaptation (Multiple)	17.79	71.25	15.20	76.41

Table 3: TER and BLEU scores on the WMT21 Wikipedia test and WMT19 IT En–De development sets.

4 Experiments

We consider the following baselines.

Do Nothing This baseline treats the original MT-generated translations as APE outputs.

w/o Adaptation (Primary baseline) An APE model trained using the curriculum training strategy without any adapters or domain-specific fine-tuning.

Transfer Learning This experiment involves updating all parameters of the APE model during the fine-tuning stage, where the data of a single domain is used.

The following experiments are conducted to assess the domain adaptation capabilities of APE systems.

Single Adapter (Dec) In this experiment, we only inject a single adapter module into the decoder blocks of the two-encoder, single-decoder transformer model. Only the adapter is updated during the fine-tuning stage, as discussed in the earlier section.

Single Adapter (Enc + Dec) Similar to the *Single Adapter (Dec)* experiment, except a single adapter module is added on top of each encoder block as well.

Huang et al. (2022) (Joint) In this experiment, instead of using a weighted combination of the adapters based on the domain classifier-generated domain distribution in the final decoder layer to generate the output, only specific domain-specific adapters are activated based on the output of the domain classifier. Unlike the Huang et al. (2022) work, we do the joint training on classification and APE task, following the training procedure discussed in subsection 3.2 as it leads to better results than separate training.

AdaptAPE The experiment uses the proposed APE domain adaptation framework.

AdaptAPE w/o Enc The experiment is similar to the *AdaptAPE* experiment, except it has no adapter modules injected into the encoder blocks. All the LLM-based experiments follow 5-shot prompting. The decoding is done using beam search with the beam width being 5.

Additionally, we conduct the following LLM-based experiments to investigate the capabilities of LLM and prompting-based techniques to perform domain-aware post-editing. The corresponding prompt templates are provided in **Appendix J**.

LLM-APE w/o Adaptation In this experiment, we prompt LLaMA-3.3-70B-Instruct to perform post-editing without any domain-specific information. The model receives only the source sentence and its raw machine translation as input.

LLM-APE w/ Adaptation (Single) This experiment follows a two-stage prompting strategy. In the first stage, the model is prompted to classify the source–translation pair into one of the predefined domains. In the second stage, the predicted domain is appended to the prompt to guide post-editing.

LLM-APE w/ Adaptation (Multiple) Similar to the single-domain setup, this experiment also

uses a two-stage prompting strategy. However, instead of selecting a single domain, the model is first prompted to estimate the similarity of the source–translation pair to each domain in the predefined list. These similarity scores are then incorporated into the prompt during the second stage to guide post-editing with cross-domain awareness.

Refer to **Appendix A** for all experimental details.

5 Results and Analysis

We use TER (Snover et al., 2006) as a primary metric to evaluate the performance of APE systems. Additionally, we also report the BLEU (Papineni et al., 2002) scores.

Tables 3, 4, and 5 summarize the results for the En–De (Indo-European), En–Mr (Indo-Aryan), and En–Ta (Indo-Dravidian) language pairs, respectively.

The experiments reveal that *Transfer Learning* is largely ineffective, particularly when the target-domain data is limited or the evaluation set is challenging, as indicated by the low TER improvement over the *Do Nothing* baseline. The model tends to overfit quickly under such conditions, resulting in marginal gains.

In contrast, *Single Adapter (Dec)* experiments show comparatively better performance, highlighting that even simple domain adaptation using decoder-side adapters can be beneficial. Further improvements observed in the *Single Adapter (Enc + Dec)* setup suggest that harmonizing the encoder and decoder during the adaptation phase enhances the model's ability to generalize to domain-specific patterns.

Substantial gains demonstrated by the *Huang et al. (2022) (Joint)* experiment over the single-adapter variants underline the effectiveness of their classifier-guided, multi-adapter approach, despite using adapters only on the decoder side.

To enable a direct comparison with their setup, we introduce the *AdaptAPE w/o Enc* variant of our model, which excludes encoder adapters. Results show that *AdaptAPE w/o Enc* outperforms or shows comparable results to *Huang et al. (2022) (Joint)*, emphasizing the advantage of using a domain classifier to compute a weighted combination of multiple adapters instead of hard selection. This also suggests that leveraging inter-domain similarities is effective, especially when domain-specific data is scarce.

Finally, the performance gains achieved by the full *AdaptAPE* model over *AdaptAPE w/o Enc* demonstrate the added value of encoder-side adapters. These improvements are likely due to better cross-lingual representation learning, which enhances both domain classification and post-edit generation.

Overall, the superior or comparable performance of *AdaptAPE* over *Huang et al. (2022) (Joint)* highlights the effectiveness of jointly training the model on both domain classification and APE tasks, as opposed to the sequential training strategy used in previous work.

While *AdaptAPE* outperforms *Huang et al. (2022) (Joint)* across all domains for En–De and En–Mr, its performance on En–Ta is more mixed: it achieves comparable results for the General domain, shows marginal gains for Health, but performs poorly on the Legal domain. One possible reason is that the Legal domain is linguistically and stylistically more divergent from General and Health, which may reduce the effectiveness of cross-domain knowledge sharing. We conjecture that when domains are highly divergent, the benefits of soft adapter combination diminish, as there is less transferable information across domains. In such scenarios, simpler methods like hard adapter selection may suffice or even outperform more complex weighting strategies.

Out-of-Domain Performance Table 6 presents the evaluation results of domain adaptation approaches on an unseen domain (Administrative) for the En–Mr and En–Ta pairs. Both language pairs include three training domains each, with General and Health being common across them. This setup allows for a controlled comparison of how domain overlap and divergence affect generalization. The results demonstrate the effectiveness of leveraging inter-domain similarities for improving performance on unseen domains. In particular, since the Legal domain (present in En–Ta) is stylistically and functionally closer to Administrative than Tourism (present in En–Mr), *AdaptAPE* achieves comparatively higher gains on the En–Ta pair, aligning well with our hypothesis.

The *AdaptAPE (LLM-as-Annotator)* rows in Tables 4 and 5 report results for models trained using the *AdaptAPE* framework, where domain labels for training data were generated using the LLaMA-3.3-70B-Instruct model. Comparing these results with those of the corresponding *AdaptAPE* experiments using human-annotated labels reveals a con-

Experiment	General		Health		Tourism	
	TER	BLEU	TER	BLEU	TER	BLEU
Do Nothing	21.89	67.50	20.14	66.81	28.22	58.26
w/o Adaptation	18.42	71.70	18.73	68.39	22.46	64.65
Transfer Learning	18.00	72.18	18.56	68.53	21.01	66.43
Single Adapter (Dec)	17.77	72.41	18.52	68.53	20.83	66.69
Single Adapter (Enc + Dec)	17.52	72.70	18.40	68.68	20.32	67.20
AdaptAPE w/o Enc	17.30	72.83	18.48	68.57	20.27	67.31
AdaptAPE	**17.21**	**72.96**	**18.25**	**68.86**	**20.06**	**67.54**
Huang et al. (2022) (Joint)	18.36	71.74	18.72	68.42	21.19	66.20
LLM-APE w/o Adaptation	25.11	63.03	31.85	53.82	34.44	50.98
LLM-APE w/ Adaptation (Single)	25.18	62.99	30.21	56.30	32.19	54.35
LLM-APE w/ Adaptation (Multiple)	25.05	63.10	29.17	57.48	30.62	57.13
AdaptAPE (LLM-as-Annotator)	17.24	72.91	19.09	67.88	22.96	64.00

Table 4: TER and BLEU scores on the WMT22 En–Mr development set. Overall Macro and Weighted TER scores are reported in Table 8.

Experiment	General		Health		Legal	
	TER	BLEU	TER	BLEU	TER	BLEU
Do Nothing	26.30	68.78	15.31	75.20	46.69	51.41
w/o Adaptation	22.22	72.87	20.40	69.62	37.71	61.47
Transfer Learning	22.00	73.13	20.63	69.37	32.45	66.88
Single Adapter (Dec)	21.01	74.25	18.86	71.35	28.74	70.63
Single Adapter (Enc + Dec)	20.56	74.85	18.8	71.44	27.20	72.21
AdaptAPE w/o Enc	19.09	76.94	17.53	72.76	26.13	73.39
AdaptAPE	18.67	**78.00**	**17.44**	**72.89**	25.52	74.11
Huang et al. (2022) (Joint)	**18.65**	77.96	17.52	72.81	**25.21**	**74.46**
LLM-APE w/o Adaptation	26.46	69.02	23.27	65.76	33.00	65.93
LLM-APE w/ Adaptation (Single)	25.91	68.39	21.04	70.39	32.48	66.54
LLM-APE w/ Adaptation (Multiple)	23.42	70.23	20.31	69.64	30.75	69.38
AdaptAPE (LLM-as-Annotator)	18.79	77.86	17.95	72.40	28.39	71.02

Table 5: TER and BLEU scores on the WMT24 En–Ta development set. Overall Macro and Weighted TER scores are reported in Table 9.

Experiment	En–Mr		En–Ta	
	TER	BLEU	TER	BLEU
Do Nothing	23.64	64.55	21.09	68.92
w/o Adaptation	20.29	68.16	19.41	71.23
AdaptAPE	**18.27**	**70.38**	**18.86**	**72.02**
Huang et al. (2022) (Joint)	18.91	69.71	20.42	70.17

Table 6: TER and BLEU scores on the out-of-domain (Administrative) evaluation set.

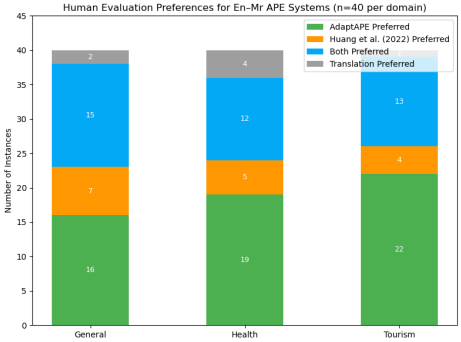

Figure 2: Human evaluation results for En–Mr APE outputs across three domains.

sistent drop in performance, particularly across non-General domains. This highlights the importance of human annotation, where domain expertise and world knowledge enable more accurate domain identification, which in turn leads to more effective domain-specific adaptation in APE systems.

Qualitative Evaluation (En–Mr) To assess whether the TER and BLEU improvements achieved by *AdaptAPE* over *Huang et al. (2022) (Joint)* translate into meaningful gains in fluency and adequacy, we conduct a human evaluation of APE outputs for the En–Mr pair. We randomly sample 40 instances from each of the three domains in the development set, resulting in 120 sentence triplets. For each instance, the source sentence, the machine translation, and the post-edits generated by both *AdaptAPE* and *Huang et al. (2022) (Joint)* are provided to a professional translator. The order of these outputs were shuffled to prevent any signal from being passed to the annotator.

The annotator was asked to perform two tasks: (1) indicate a preference considering the domain between the two post-edited outputs or the original translation, with the option to mark both APE outputs as equally good; and (2) flag any instances where a post-edit introduces an overcorrection, i.e., an unnecessary change (not necessarily only those that adversely affect the translation quality).

Figure 2 presents the results of the human preference-based evaluation. We observe a consistent preference for *AdaptAPE* across all three domains, with the largest margin in the Health and Tourism domains. In contrast, the outputs of *Huang et al. (2022) (Joint)* are preferred relatively less often, and in only a small number of cases is the original translation favored. The low frequency of *"Huang et al. (2022) (Joint)"* and a high frequency of "both preferred" selections together indicates that while both systems produce competitive outputs, *AdaptAPE* has a clear overall advantage.

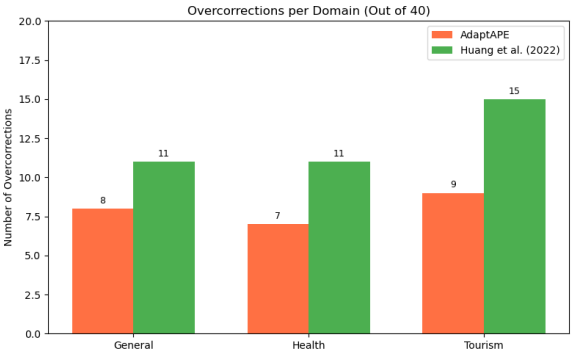

Figure 3: Number of instances exhibiting overcorrection.

Figure 3 reports the number of instances where a human annotator marked a post-edit as exhibiting overcorrection i. e., unnecessary edits to translations. Across all domains, *AdaptAPE* introduces fewer overcorrections compared to the baseline system from *Huang et al. (2022) (Joint)*. The difference is especially notable in the Tourism domain, where the baseline introduces 15 overcorrections out of 40 instances, whereas *AdaptAPE* limits this to just 9. This indicates that *AdaptAPE*'s edits are generally more conservative and contextually appropriate, aligning with its goal of minimal yet effective post-editing. Two En–Mr examples are discussed in **Appendix I**.

Kindly refer to Appendix C for analysis on sensitivity of APE performance on the classifier performance. We discuss the impact of joint and separate training on classification and APE tasks in Appendix E. The impact of adapter module injection to encoders is analyzed in Appendix F. An example shocasing how soft weighing of adapters impact the APE output is presented in Appendix G In Appendix K, we discuss the comparison with the LLM-based experiments.

6 Conclusion and Future Work

We introduced *AdaptAPE*, a domain-adaptation APE framework. Unlike prior approaches that

rely on hard adapter domain selection, *AdaptAPE* uses soft selection, a classifier-guided weighted combination of domain-specific adapters, enabling soft, flexible, and cross-domain knowledge integration during training and inference. Our experiments across three linguistically diverse language pairs, English–German, English–Marathi, and English–Tamil, demonstrate that *AdaptAPE* achieves consistent gains across domains and varied experimental settings. We further show that our approach generalizes well to out-of-domain settings by leveraging cross-domain similarities, and outperforms classifier-guided hard adapter selection approaches. Moreover, by comparing with large LLM-based APE systems, we highlight the continued relevance and effectiveness of compact, specialized Transformer-based models for domain-aware post-editing, particularly in low-resource settings. We release human-annotated domain labels for En–Hi, En–Mr, and En–Ta WMT APE datasets under a CC BY-NC license to support further domain-aware APE research.

While this work investigates the effectiveness of classifier-guided domain adaptation for high-precision tasks like APE, identifying which domain adaptation strategies are most effective across different settings remains an important direction for future research. In addition, future work can explore the coupling between domain-adapted Quality Estimation (QE) and APE. Our findings suggest that domain-aware APE is a promising step toward deploying reliable, high-quality post-editing systems in real-world, mixed-domain applications.

7 Limitations

A primary limitation of the proposed approach is its reliance on prior knowledge of the number of domains and access to domain labels for each APE triplet in the training data. This requirement restricts the applicability of the model in settings where domain labels are unavailable or noisy. Due to the limited availability of labeled APE data across diverse domains, we have evaluated the model's robustness to out-of-domain inputs using synthetic APE data. Additionally, on the data annotation front, each triplet in our dataset has been labeled with a single domain, although we recognize that sentences often exhibit characteristics of multiple domains. Incorporating multi-domain annotations per triplet could provide a richer training signal and lead to more nuanced domain adaptation.

8 Ethics Statement

Our APE models are developed using publicly accessible datasets cited in this paper. Apart from adding a domain label to each sample in the datasets, we do not perform generation or collection of data as a part of this work. The professional translators who annotated the APE triplets with domain labels are paid as per the industrial standards. We have their consent to publicly release these domain labels with this work. Additionally, these datasets serve as standard training datasets or benchmarks introduced in recent WMT shared tasks. The datasets do not contain any user information, ensuring the privacy and anonymity of individuals. We acknowledge that all datasets carry inherent biases, and as a result, computational models are bound to acquire biased information from them.

9 CO2 Emission Related to Experiments

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 $kgCO_2eq/kWh$. A cumulative of 100 hours of computation was performed on hardware of type A100 PCIe 40/80GB (TDP of 250W).

Total emissions are estimated to be 10.8 $kgCO_2eq$ of which 0 percents were directly offset.

Estimations were conducted using the Machine-Learning Impact calculator presented in (Lacoste et al., 2019).

References

[Akhbardeh et al.2021] Akhbardeh, Farhad, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In Barrault, Loic, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre

Martins, Makoto Morishita, and Christof Monz, editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online, November. Association for Computational Linguistics.

[Bansal et al.2013] Bansal, Akanksha, Esha Banerjee, and Girish Nath Jha. 2013. Corpora creation for indian language technologies–the ilci project. In *the sixth Proceedings of Language Technology Conference (LTC '13)*.

[Bapna and Firat2019] Bapna, Ankur and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In Inui, Kentaro, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China, November. Association for Computational Linguistics.

[Bhattacharyya et al.2022] Bhattacharyya, Pushpak, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2022. Findings of the WMT 2022 shared task on automatic post-editing. In Koehn, Philipp, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 109–117, Abu Dhabi, United Arab Emirates (Hybrid), December. Association for Computational Linguistics.

[Bhattacharyya et al.2023] Bhattacharyya, Pushpak, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2023. Findings of the WMT 2023 shared task on automatic post-editing. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 672–681, Singapore, December. Association for Computational Linguistics.

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

[Chatterjee et al.2018] Chatterjee, Rajen, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels, October. Association for Computational Linguistics.

[Chatterjee et al.2019] Chatterjee, Rajen, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy, August. Association for Computational Linguistics.

[Chatterjee et al.2020] Chatterjee, Rajen, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In Barrault, Loïc, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online, November. Association for Computational Linguistics.

[Deoghare et al.2023] Deoghare, Sourabh, Diptesh Kanojia, Fred Blain, Tharindu Ranasinghe, and Pushpak Bhattacharyya. 2023. Quality estimation-assisted automatic post-editing. In Bouamor, Houda, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1686–1698, Singapore, December. Association for Computational Linguistics.

[Deoghare et al.2024] Deoghare, Sourabh, Diptesh Kanojia, and Pushpak Bhattacharyya. 2024. Together we can: Multilingual automatic post-editing for low-resource languages. In Al-Onaizan, Yaser, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10800–10812, Miami, Florida, USA, November. Association for Computational Linguistics.

[Feng et al.2022] Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In

Muresan, Smaranda, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May. Association for Computational Linguistics.

[Huang et al.2022] Huang, Xiaoying, Xingrui Lou, Fan Zhang, and Tu Mei. 2022. LUL's WMT22 automatic post-editing shared task submission. In Koehn, Philipp, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 689–693, Abu Dhabi, United Arab Emirates (Hybrid), December. Association for Computational Linguistics.

[Lacoste et al.2019] Lacoste, Alexandre, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

[Lee et al.2020] Lee, Jihyung, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020. POSTECH-ETRI's submission to the WMT2020 APE shared task: Automatic post-editing with cross-lingual language model. In Barrault, Loïc, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 777–782, Online, November. Association for Computational Linguistics.

[Oh et al.2021] Oh, Shinhyeok, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In Barrault, Loic, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online, November. Association for Computational Linguistics.

[Papineni et al.2002] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Isabelle, Pierre, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

[Ramesh et al.2022] Ramesh, Gowtham, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

[Sharma et al.2021] Sharma, Abhishek, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In Barrault, Loic, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors, *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online, November. Association for Computational Linguistics.

[Snover et al.2006] Snover, Matthew, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12. Association for Machine Translation in the Americas.

[Zerva et al.2024] Zerva, Chrysoula, Frederic Blain, José G. C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. 2024. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In Haddow, Barry, Tom Kocmi, Philipp Koehn, and Christof Monz, editors, *Proceedings of the Ninth Conference on Machine Translation*, pages 82–109, Miami, Florida, USA, November. Association for Computational Linguistics.

A Experimental Details

The section gives details for the experiments discussed in Section 4. Our two-encoder, single-decoder transformer-based APE architecture consists of 6 blocks in each encoder and decoder. We use the hidden size of 512, the feedforward network dimension is 2048, and each multi-head attention

has 8 heads. We set the residual dropout probability to 0.1 and use a label smoothing value of 0.1. Our APE models were trained using a batch size of 32 for up to 1000 epochs, with early stopping activated if validation performance did not improve for 5 consecutive steps. The Adam optimizer was employed with a learning rate of 5 x 10^{-5}, and momentum parameters β_1 equals 0.9, and β_2 equals 0.997. We included a linear warm-up schedule with 25,000 warm-up steps. For decoding, beam search with a beam width of 5 was used. Early stopping with a patience of 10 validation steps was adopted for experiments. All training was performed on NVIDIA A100 GPUs. The APE model consists of roughly 40 million parameters. End-to-end training, including the domain-adaptation phase with curriculum learning, took approximately 48 hours. For preprocessing, English and German texts were handled using the NLTK library[2], while Marathi and Tamil data were processed with the IndicNLP toolkit[3]. All training and inference were implemented in PyTorch[4]. TER scores were computed using the sacrebleu[5] library.

For 5-shot prompting of LLaMA-3.3-70B-Instruct in the single-domain classification setting, we randomly select five examples from the respective development set, ensuring that at least one example from each target domain is included. This helps the model ground its predictions with clear, domain-specific cues. For multi-domain classification, we construct the 5-shot example pool from development set instances where the machine translation and reference post-edit have a Translation Edit Rate (TER) score below 15. This filtering helps ensure a high-quality signal for domain association. Additionally, we stratify the pool by translation length, selecting at least one example each from three length buckets: <10 tokens, 10–20 tokens, and >20 tokens, to promote robustness across sentence lengths. The same pool is used for domain-unaware post-editing and multi-domain-based post-editing. For the single-domain-based post-editing, we ensure we pick at least one example of each domain, too.

[2]https://www.nltk.org/

[3]https://github.com/anoopkunchukuttan/indic_nlp_library

[4]https://pytorch.org/

[5]https://github.com/mjpost/sacrebleu

B Datasets

This section extends the discussion of datasets started in Subsection 3.3. For our experiments, we utilize datasets from the WMT19 (Chatterjee et al., 2019), WMT21 (Akhbardeh et al., 2021) APE datasets for En–De experiments.

WMT24[6], and WMT22 (Bhattacharyya et al., 2022) APE shared tasks for English–German. The synthetic data contains around 7M APE triplets, while the real APE sets contain about 13K IT and 7K Wikipedia domain triplets. WMT22 (Bhattacharyya et al., 2022) En–Mr APE data used in the study contains 2.5M and 18K synthetic and real APE triplets, respectively. For the En–Ta pair, we use 2.5M and 7K synthetic and real APE triplets released by WMT24 QEAPE shared task (Zerva et al., 2024). For all pairs, about 1K triplets each are present in the corresponding development or test datasets.

To support APE training, we leverage parallel corpora for each language pair. For English–Marathi and English–Tamil, we use data sourced from the Anuvaad[7], Samanantar (Ramesh et al., 2022), and ILCI (Bansal et al., 2013) corpora, each comprising roughly 6M and 4M sentence pairs, respectively. For English–German, we make use of the News Commentary dataset provided as part of the WMT22 machine translation task, which includes approximately 10M sentence pairs.

C Sensitivity to Classifier Performance

To analyze the sensitivity of AdaptAPE to errors in the domain classifier, we introduced controlled perturbations to the classifier's softmax outputs. Specifically, we applied two types of perturbations: (i) **uniform noise**, where the original probability distribution p is blended with the uniform distribution u at different levels according to $p' = (1-\alpha)p + \alpha u$, with $\alpha \in 0, 0.33, 0.66, 1.0$, and (ii) **+ shifts and rotations**, where in addition to uniform noise, the probability mass is randomly reassigned across classes based on random left or right shift by one or two positions, simulating cases where the classifier not only becomes uncertain but also misidentifies the dominant class with certainty too. Results of this experiment for the En–Mr pair are compiled in Table 7. We can see that TER increases consistently as uniform noise (α) increases, and the effect is amplified when shifts and rotations

[6]WMT24 QEAPE Shared Subtask

[7]Anuvaad Parallel Corpus

Alpha/Scenario	General	Health	Tourism
0	17.21	18.25	20.06
0.33	17.35	18.48	20.40
0.66	17.60	18.80	21.00
1	18.83	19.10	21.59
0.66 + shifts + rotations	17.86 ± 0.10	19.05 ± 0.12	21.46 ± 0.15
0.33 + shifts + rotations	18.10 ± 0.18	19.34 ± 0.10	21.70 ± 0.22
0.0 + shifts + rotations	18.23 ± 0.26	19.38 ± 0.19	22.51 ± 0.34

Table 7: TER scores on the evaluation set by En–Mr AdaptAPE with classification noise injection.

are introduced, with stronger degradation at lower α values. Reduced classifier confidence alone leads to performance loss, but redistribution of probability mass across incorrect domains makes the system more vulnerable.

D Overall Performance on En–Mr and En–Ta Pairs

Tables 8 and 9 show the overall performance of APE models in terms of TER scores on En–Mr and En–Ta pairs.

E Joint vs Separate Training of Adapters and Classifier

To assess the significance of the joint training of adapters and the classifier, we report a variant, *AdaptAPE (separate training)*, where the training paradigm differs from the joint multitask setup. The first two steps remain identical to AdaptAPE until the additional components are introduced. After this point, we first injected and trained the encoder adapters together with the domain classifier while keeping all other parameters frozen. In the next stage, we froze both the encoder adapters and classifier, and then injected and trained only the decoder adapters for the APE task. Tables 10 and 11 given below, show classifier and APE performances for En–Mr and En–Ta pairs, respectively.

As seen in these tables, joint training consistently yields higher average macro and weighted F1 scores, while maintaining or lowering TER across domains. Separate training, in contrast, weakens classifier performance and increases TER. On a side note, the comparison further shows that our soft-selection strategy continues to outperform hard selection, reinforcing its effectiveness across domains. We will include a detailed analysis and discussion in the paper.

F Impact of Encoder Adapter on Classification and APE Performance

We compare AdaptAPE with and without encoder adapters for En–Mr and En–Ta pairs and analyze the classifier and APE performance to see whether and how the addition of an adapter module on the encoder side helps. Tables 12 and 13 show the results of AdaptAPE with *(AdaptAPE with Enc)* and without *(AdaptAPE w/o Enc)* encoder adapter experiments for En–Mr and En–Ta pairs. The results show that removing encoder adapters leads to a consistent drop in classifier performance, along with slightly worse TER, underscoring the benefit of adding them.

G Example Analysis: Effect of Soft Adapter Weighting on En–Mr Translation

We discuss an En–Mr example shown in Figure 4, which shows how soft selection has led to qualitatively better output compared to hard adapter selection. The example highlights how different approaches handle domain overlap in En–Mr APE. The raw MT translation takes a literal path, rendering visitors as 'paryatak' (general tourists), translating district authorities literally as 'jeelha adheekaaryaanni,' omitting 'valid,' and softening the requirement into a polite 'veenanti keli aahe' ('requested'), thereby missing both domain-specific nuances.

The *w/o Adaptation* model strengthens the mandate by changing the translation of 'requested' to 'bandhankaarak kele aahe' ('made mandatory') and introduces 'saadhaar' as a correct but less common rendering of valid.

The hard-selection Tourism adapter ((Huang et al., 2022)) makes domain-specific improvements: 'narmadaa pareekramaa' for 'Narmada river pilgrimage' and 'yaatreekaru' for visitors, situating

Experiment	General	Health	Tourism	Macro TER	Weighted TER
Do Nothing	21.89	20.14	28.22	23.42	23.33
w/o Adaptation	18.42	18.73	22.46	19.87	19.79
Transfer Learning	18.00	18.56	21.01	19.19	19.13
Single Adapter (Dec)	17.77	18.52	20.83	19.04	18.97
Single Adapter (Enc + Dec)	17.52	18.40	20.32	18.75	18.68
AdaptAPE w/o Enc	17.30	18.48	20.27	18.68	18.61
AdaptAPE	**17.21**	**18.25**	**20.06**	**18.51**	**18.44**
(Huang et al., 2022) (Joint)	18.36	18.72	21.19	19.42	19.37
LLM-APE w/o Adaptation	25.11	31.85	34.44	30.47	30.19
LLM-APE w/ Adaptation (Single)	25.18	30.21	32.19	29.19	28.98
LLM-APE w/ Adaptation (Multiple)	25.05	29.17	30.62	28.28	28.11
AdaptAPE (LLM-as-Annotator)	17.24	19.09	22.96	19.76	19.63

Table 8: TER scores across domains and training configurations for the En–Mr APE task.

Experiment	General	Health	Legal	Macro TER	Weighted TER
Do Nothing	26.30	15.31	46.69	29.43	28.65
w/o Adaptation	22.22	20.40	37.71	26.78	25.64
Transfer Learning	22.00	20.63	32.45	25.03	24.27
Single Adapter (Dec)	21.01	18.86	28.74	22.87	22.40
Single Adapter (Enc + Dec)	20.56	18.80	27.20	22.19	21.78
AdaptAPE w/o Enc	19.09	17.53	26.13	20.92	20.46
AdaptAPE	**18.67**	**17.44**	25.52	**20.54**	**20.08**
(Huang et al., 2022) (Joint)	18.65	17.52	**25.21**	20.46	20.01
LLM-APE w/o Adaptation	26.46	23.27	33.00	27.58	27.30
LLM-APE w/ Adaptation (Single)	25.91	21.04	32.48	26.48	26.34
LLM-APE w/ Adaptation (Multiple)	23.42	20.31	30.75	24.83	24.48
AdaptAPE (LLM-as-Annotator)	18.79	17.95	28.39	21.71	20.98

Table 9: TER scores across domains and training configurations for the En–Ta APE task.

Experiment	Gen (F1)	Health (F1)	Tour (F1)	Macro F1	Weighted F1	Gen (TER)	Health (TER)	Tour (TER)
AdaptAPE (joint)	79.5	**83.8**	**82.6**	**81.97**	**81.84**	**17.21**	**18.25**	**20.06**
AdaptAPE (separate)	**81.4**	76.7	77.9	78.67	78.81	17.66	18.83	20.94

Table 10: Comparison of joint vs separate AdaptAPE training across domains on En–Mr APE.

Experiment	Gen (F1)	Health (F1)	Legal (F1)	Macro F1	Weighted F1	Gen (TER)	Health (TER)	Legal (TER)
AdaptAPE (joint)	74.7	**80.4**	**88.1**	**81.07**	**79.47**	**18.67**	**17.44**	**25.52**
AdaptAPE (separate)	**72.6**	75.1	84.4	77.37	76.17	19.26	18.02	26.39

Table 11: Comparison of joint vs separate AdaptAPE training across domains on En–Ta APE.

Experiment	Gen (F1)	Health (F1)	Tour (F1)	Macro F1	Weighted F1	Gen (TER)	Health (TER)	Tour (TER)
AdaptAPE (with Enc)	**79.5**	**83.8**	**82.6**	**81.97**	**81.84**	**17.21**	**18.25**	**20.06**
AdaptAPE (w/o Enc)	77.9	82.1	81.0	80.35	80.23	17.30	18.48	20.27

Table 12: Comparison of AdaptAPE with and without encoder on En–Mr APE.

Experiment	Gen (F1)	Health (F1)	Legal (F1)	Macro F1	Weighted F1	Gen (TER)	Health (TER)	Legal (TER)
AdaptAPE (with Enc)	**74.7**	**80.4**	**88.1**	**81.07**	**79.47**	**18.67**	**17.44**	**25.52**
AdaptAPE (w/o Enc)	74.3	79.7	87.2	80.40	78.88	19.09	17.53	26.13

Table 13: En–Ta classifier and APE performance with and without using encoder adapters.

English-Marathi Example			
Source	Visitors undertaking the Narmada river pilgrimage are required by the district authorities to carry valid vaccination certificates.		
Translation	नर्मदा नदीची यात्रा करणाऱ्या पर्यटकांना जिल्हा अधिकाऱ्यांनी लसीकरण प्रमाणपत्र आणण्याची विनंती केली आहे.	Narmadaa (Narmada) nadichee (of river) yaatraa (pilgimage) karanyaaryaa (those who are doing) paryatakaannaa (tourists) jeelhaa (district) adheekaaryaanni (by officers) lasikaran (vaccination) pramaanapatra (certificate) aananyaachi (of bringing) veenanti (request) keli (done) aahe (is).	The tourists undertaking the Narmada river pilgrimage have been requested by the district officers to carry vaccination certificates.
w/o Adaptation	नर्मदा नदीची यात्रा करणाऱ्या पर्यटकांना जिल्हा अधिकाऱ्यांनी साधार लसीकरण प्रमाणपत्र आणणे बंधनकारक केले आहे.	Narmadaa (Narmada) nadichee (of river) yaatraa (pilgrimage) karanyaaryaa (those who are doing) paryatakaannaa (tourists) jeelhaa (district) adheekaaryaanni (by officers) saadhaar (with evidence) lasikaran (vaccination) pramaanapatra (certificate) aanane (to bring) bandhankaarak (mandatory) kele (done) aahe (is).	The tourists undertaking the Narmada river pilgrimage have been required by the district officers to carry a valid vaccination certificate.
Huang et al. (2022) (Joint) Hard Selection (Tourism)	नर्मदा परिक्रमा करणाऱ्या यात्रिकरुंना जिल्हा अधिकाऱ्यांनी वैध लसीकरण प्रमाणपत्र आणणे बंधनकारक केले आहे.	Narmadaa (Narmada) pareekramaa (circumambulation) karanyaaryaa (those who are doing) yaatreekarunna (to pilgrims) jeelhaa (district) adheekaaryaanni (by officers) vaidh (valid) lasikaran (vaccination) pramaanapatra (certificate) aanane (to bring) bandhankaarak (mandatory) kele (done) aahe (is).	The pilgrims performing the Narmada circumambulation have been required by the district officers to carry a valid vaccination certificate.
AdaptAPE (General: 0.24, Health: 0.18, Tourism: 0.58)	नर्मदा परिक्रमा करणाऱ्या यात्रिकरुंना जिल्हा प्रशासनाने वैध लसीकरण प्रमाणपत्र आणणे अनिवार्य केले आहे.	Narmadaa (Narmada) nadichee (of river) pareekramaa (circumambulation) karanyaaryaa (those who are doing) yaatreekarunna (to pilgrims) jeelhaa (district) prashaasanaane (by administration) vaidh (valid) lasikaran (vaccination) pramaanapatra (certificate) aanane (to bring) bandhankaarak (mandatory) kele (done) aahe (is).	The pilgrims performing the Narmada circumambulation have been required by the district administration to carry a valid vaccination certificate.

Figure 4: An En–Mr example showing the effect of soft adapter weighting.

the translation in the religious/spiritual tourism context.

AdaptAPE with soft weights (General: 0.24, Health: 0.18, Tourism: 0.58) combines signals from all three domains, and we see further improvements to the translation. The higher Tourism weight preserves 'narmadaa pareekramaa' and 'yaatreekaru,' the Health contribution ensures 'vaidh lasikaran pramaanapatra' ('valid vaccination certificate'), and the General weight refines district authorities into the more fluent 'jeelhaa prashaasan.' The final output is both more accurate and more natural, demonstrating that AdaptAPE's weighted soft combination successfully aggregates cross-domain knowledge, unlike hard-selection methods that privilege only one domain.

H Misclassified Examples

This section discusses two representative misclassified examples in connection with the LLM-based domain-labeling discussed in Subsection 3.3.

Figure 5 shows example instances that have been misclassified by LLM. The LLM misclassified the first sentence pair as General, possibly due to its factual tone and absence of explicit tourism-related keywords, despite mention of a location name and the mention of a community. In contrast, the human annotator correctly identified it as Tourism, recognizing that references to traditional instruments and local communities are often part of cultural tourism narratives.

Similarly, in case of second example, despite referencing "rnib" (Royal National Institute for Blind People), the LLM misclassified the pair as General. This could be due to the lowercase form of the abbreviation and the absence of a translated equivalent in the Marathi sentence, which weakens the health-related cue. In contrast, a human annotator recognized "rnib" as a health institution and infered the domain correctly based on world knowledge.

These representative examples underscore the importance of human annotation in domain labeling, particularly for cases where domain cues are subtle, implicit, or lost in translation.

I Qualitative Examples

Figure 6 shows two qualitative examples with post-edits from the three APE systems. In the first example, among the four outputs, *AdaptAPE* provides the most accurate and domain-appropriate translation. Unlike the baseline translation and the *w/o Adaptation* output, which retain vague or generic terms like 'phod' (boil/blister), *AdaptAPE* correctly uses the medical term for abscess, reflecting a better understanding of domain-specific terminology. Additionally, it preserves the formal structure and accurately captures the translation of the 'sample of pus' phrase. The post-edit by (Huang et al., 2022) does capture 'sample of pus' but fails to use a medically accurate term for abscess and slightly simplifies the phrasing.

The second example illustrates how *AdaptAPE* provides a more fluent and semantically faithful post-edit by capturing the nuance of the source sentence, which could be due to help from General domain. Unlike the baseline and *w/o Adaptation* outputs, which contain incorrect terminology such as *gemini kawa* and lack structural depth, *AdaptAPE* accurately uses *'gamini kavyachā vāpar karūn'* and reconstructs the sentence, closely reflecting the intent of the original English sentence.

J Prompt Templates

Tables 14 to 18 show prompt templates used for prompting the LLM-based experiments discussed in Section 4.

English-Marathi Misclassified Examples		
Source	An 'S' shaped trumpet, this instrument is mostly used by the 'Sargara' community of Rajasthan.	
Translation	'एस' आकाराची ट्रम्पेट, हे वाद्य राजस्थानमधील 'सरगारा' समुदायाद्वारे प्रामुख्याने वापरले जाते.	'S' aakaaraachi *('S' shaped)* trampet *(trumpet)*, he *(this)* waadya *(instrument)* raajasthaanamadhil *(of Rajasthan)* 'Sargaaraa' *(Sargara)* samudaayaadwaare *(by the community)* praamukhyaane *(mainly)* waaparale *(being used)* jaate *(is)*.
Human Annotated Domain	Tourism	
LLM Annotated Domain	General	
Source	The rnib's helpline is open monday to friday from 8am to 8pm and saturday from 9am to 1pm.	
Translation	ही हेल्पलाईन सोमवारी सकाळी 8 ते रात्री 8 आणि शनिवारी सकाळी 9 ते दुपारी 1 वाजेपर्यंत खुली आहे.	Hi *(This)* helplaayin *(helpline)* somawaari (on monday) sakaali (morning) 8 te (to) raatri (at night) 8 aanee (and) shaneewaari (on saturday) sakaali (morning) 9 te (to) doopaari (afternoon) 1 waajeparyant (till time) khooli (open) aahe (is).
Human Annotated Domain	Health	
LLM Annotated Domain	General	

Figure 5: Misclassified En–Mr examples. The examples are misclassified by LLaMA 3.3-70B-Instruct as General domain.

Prompt Template for Single-Domain Classification
You are a domain classification expert. Given a parallel sentence pair in English and <Target Language>, classify it into one of the following domains: Domains: [<Domain 1>, <Domain 2>, <Domain 3>] Instructions: - Choose one domain that best fits the content of the sentence pair. - If the sentence appears generic but fits naturally in a specialized domain, assign that domain. - Answer with only the domain name. Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain: <Domain i> . . . . . Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain: <Domain j> Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain:

Table 14: Prompt used to instruct LLaMA-3.3-70B-Instruct for single-domain classification of sentence pairs.

K Comparison with LLM-based Approaches

Tables 3, 4, and 5 show that LLM-based approaches fail to surpass the performance of *AdaptAPE* on the En–De pair, despite German being a high-resource language. This underscores the inherent difficulty of the APE task, where high-precision edits are required over already high-quality translations. On the En–Mr and En–Ta pairs, the performance of LLM-based approaches is notably worse, likely due to the comparatively limited representation of Marathi and Tamil in the LLM's pretraining corpus.

However, the results from the *LLM-APE w/ Adaptation (Single)* setup consistently outperform the *LLM-APE w/o Adaptation* baseline on both domains of En–De, and on all but the General domain for En–Mr and En–Ta. This highlights the utility of domain-aware prompting in improving post-editing performance. Furthermore, the additional gains observed in the *LLM-APE w/ Adaptation (Multiple)* setting emphasize the value of leveraging cross-domain similarities, which is the central idea of our proposed *AdaptAPE* framework.

English-Marathi Examples			
Source	If an abscess is found, a procedure known as CT-guided aspiration may be used to remove a sample of pus for testing.		
Translation	जर फोड आढळली तर सीटी-गाइडेड एस्पिरेशन म्हणून ओळखली जाणारी प्रक्रिया चाचणीसाठी मच्छीचे नमुने काढून टाकण्यासाठी वापरली जाऊ शकते.	Jar (if) phod (abscess) aadhalali (is found) tara (then) citi-gaayided (CT-guided) espireshan (aspiration) mhanun (as) olakhali (known) jaanaari (being) prakreeyaa (procedure) chaachanisaathi (for testing) machchiche (pus) namoone (sample) kaadhun (by removing) taakanyaasaathi (for discarding/removing) waaparali (used) jaau (can be) shakate (done).	If an abscess is found, a procedure known as CT-guided aspiration may be used to remove and discard samples of fish for testing.
w/o Adaptation	जर फोड आढळली, तर सीटी-गाइडेड एस्पिरेशन म्हणून ओळखली जाणारी प्रक्रिया चाचणीसाठी नमुना काढण्यासाठी वापरली जाऊ शकते.	Jar (if) phod (abscess) aadhalali (is found) tara (then) citi-gaayided (CT-guided) espireshan (aspiration) mhanun (as) olakhali (known) jaanaari (being) prakriyaa (procedure) chaachanisaathi (for testing) namoona (sample) kaadhanyaasaathi (for removing) vaaparali (used) jaau (can be) shakate (done).	If an abscess is found, a procedure known as CT-guided aspiration may be used to extract a sample for testing.
Huang et al. (2022) (Joint)	जर फोड आढळली, तर सीटी-गाइडेड एस्पिरेशन ही प्रक्रिया चाचणीसाठी पूचा नमुना काढण्यासाठी वापरली जाऊ शकते.	Jar (if) phod (abscess) aadhalali (is found) tar (then) citi-gaayided (CT-guided) espireshan (aspiration) hi (this) prakriyaa (procedure) chaachanisaathi (for testing) poocha (of pus) namoona (sample) kaadhanyaasaathi (for removing) vaaparali (used) jaau (can be) shakate (done).	If an abscess is found, this procedure called CT-guided aspiration may be used to extract a sample of pus for testing.
AdaptAPE	जर सपुयक्षत आढळला, तर 'सीटी-गाइडेड एस्पिरेशन' म्हणून ओळखली जाणारी प्रक्रिया चाचणीसाठी पूचा नमुना काढण्यासाठी वापरली जाऊ शकते.	Jar (if) sapuyakshat (abscess) aadhalala (is found) tar (then) 'citi-guided (CT-guided) aspiration' mhanun (as) olakhali (known) jaanaari (being) prakriyaa (procedure) chaachanisaathi (for testing) poocha (of pus) namoona (sample) kaadhanyaasaathi (for extracting) vaaparali (used) jaau (can be) shakate (done).	If an abscess is found, a procedure known as 'CT-guided aspiration' may be used to extract a sample of pus for testing.
Comment	'*Phod*' is a general word in Marathi which is more commonly used to denote blisters. AdaptAPE uses the correct translation 'sapuyakshat' for abscess.		
Source	Visitors to Pratapgad Fort often learn how Shivaji Maharaj skillfully used Gamini Kawa (guerrilla tactics) to defeat powerful enemies.		
Translation	प्रतापगड किल्ल्यावर येणारे पर्यटक शिकतात की शिवाजी महाराजांनी गेमिनी कावा वापरून शक्तिशाली शत्रूंना पराभूत केले.	Prataapagad (Pratapgad) Killyaawar (at the fort) yenaare (visiting) paryatak (tourists) sheekataat (learn) ki (that) sheevaaji (Shivaji) mahaaraajaani (by Maharaj) gemini (Gamini) kaawaa (tactic) waaparun (by using) shakteeshaali (powerful) shatrunna (enemies) paraabhut (defeated) kele (did).	Tourists visiting Pratapgad Fort learn that Shivaji Maharaj defeated powerful enemies using Gemini Kawa.
w/o Adaptation	प्रतापगड किल्ल्यावर येणारे पर्यटक शिकतात की शिवाजी महाराजांनी गेमिनी कावा वापरून बलवान शत्रूंना पराभूत केले.	Prataapagad (Pratapgad) Killyaawar (at the fort) yenaare (visiting) paryatak (tourists) sheekataat (learn) ki (that) sheevaaji (Shivaji) mahaaraajaani (by Maharaj) gemini (Gamini) kaawaa (tactic) waaparun (by using) balawaan (strong) shatrunna (enemies) paraabhut (defeated) kele (did).	Tourists visiting Pratapgad Fort learn that Shivaji Maharaj defeated strong enemies using Gamini Kawa.
Huang et al. (2022) (Joint)	प्रतापगड किल्ल्यावर येणारे पर्यटक हे शिकतात की शिवाजी महाराजांनी गामिनी कावा वापरून शक्तिशाली शत्रूंना हरवले.	Prataapagad (Pratapgad) Killyaawar (at the fort) yenaare (visiting) paryatak (tourists) he (they) sheekataat (learn) ki (that) sheevaaji (Shivaji) mahaaraajaani (by Maharaj) gaamini (Gamini) kaawaa (tactic) waaparun (by using) shakteeshaali (powerful) shatrunna (enemies) harawale (defeated).	Tourists visiting Pratapgad Fort learn that Shivaji Maharaj used Gamini Kawa to defeat powerful enemies.
AdaptAPE	प्रतापगड किल्ल्यावर येणारे पर्यटक सहसा शिवाजी महाराजांनी गमिनी काव्याचा वापर करून शक्तिशाली शत्रूंना कसे पराभूत केले हे शिकतात.	Prataapagad (Pratapgad) Killyaawar (at the fort) yenaare (visiting) paryatak (tourists) sahasaa (generally / typically) sheevaaji (Shivaji) mahaaraajaani (by Maharaj) gamini (guerrilla) kaawyaachaa (of tactics) waapar (use) karun (by doing) shakteeshaali (powerful) shatrunnaa (enemies) kase (how) parabhut (defeated) kele (did) he (this) shikatat (learn).	Tourists visiting Pratapgad Fort typically learn how Shivaji Maharaj used Gamini Kawa to defeat powerful enemies.
Comment	AdaptAPE correctly translates 'often' as 'sahasaa,' correctly spells 'Gamini Kawa' in Marathi, Significantly improves fluency.		

Figure 6: En–Mr examples showing comparison between *w/o Adaptation*, *(Huang et al., 2022)*, and *AdaptAPE*.

Prompt Template for Multi-Domain Classification
You are a domain classification expert. Given a parallel sentence pair in English and <Target Language>, assess how closely the sentence pair relates to each domain from a predefined domain list. Output a score from 0 to 1 for each domain, where higher scores indicate higher relevance. Domains: [<Domain 1>, <Domain 2>, <Domain 3>] Instructions: - The scores must sum to 1. - Use your best judgment based on meaning, terminology, and context. Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution: <Domain 1>: <Value 11>, <Domain 2>: <Value 12>, <Domain 3>: <Value 13> . . . . . Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution: <Domain 1>: <Value 51>, <Domain 2>: <Value 52>, <Domain 3>: <Value 53> Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution:

Table 15: Prompt used to instruct LLaMA-3.3-70B-Instruct for multi-domain classification of sentence pairs.

Prompt Template for Post-Editing Without Domain Information
You are a professional post-editor specializing in translation correction. Given an English source sentence, its machine-translated <Target Language> version, and the domain, your task is to post-edit the <Target Language> translation to make it fluent, accurate. Instructions: - Maintain the meaning of the source sentence, preserve the terminology and style. - Make minimal edits to post-edit the translation. Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Post-Edit: <Post-Edited Sentence> . . . . . Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Post-Edit: <Post-Edited Sentence> Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Post-Edit:

Table 16: Prompt used to instruct LLaMA-3.3-70B-Instruct for performing post-editing without using any domain information.

Prompt Template for Post-Editing Using a Single-domain Label
You are a professional post-editor specializing in domain-specific translation correction. Given an English source sentence, its machine-translated <Target Language> version, and the domain, your task is to post-edit the <Target Language> translation to make it fluent, accurate, and appropriate for the given domain. Instructions: - Maintain the meaning of the source sentence, preserve domain-specific terminology and style. - Make minimal edits to post-edit the translation. Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain: <Domain i> Post-Edit: <Post-Edited Sentence> . . . . . Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain: <Domain k> Post-Edit: <Post-Edited Sentence> Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain: <Domain j> Post-Edit:

Table 17: Prompt used to instruct LLaMA-3.3-70B-Instruct for performing post-editing using a single-domain label.

Prompt Template for Post-Editing Using a Domain Distribution
You are a professional post-editor specializing in domain-specific translation correction. Given an English source sentence, its machine-translated <Target Language> version, and the domain distribution, your task is to post-edit the <Target Language> translation to make it fluent, accurate, and appropriate for the given domain. Instructions: - A domain distribution indicates the relevance of each domain to the sentence pair. - Use the domain distribution to guide your edits, focusing more on the higher-weighted domains while still preserving the meaning of the source. - Maintain the meaning of the source sentence, preserve domain-specific terminology and style. - Make minimal edits to post-edit the translation. Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution: <Domain 1>: <Value 11>, <Domain 2>: <Value 12>, <Domain 3>: <Value 13> Post-Edit: <Post-Edited Sentence> . . . . . Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution: <Domain 1>: <Value 51>, <Domain 2>: <Value 52>, <Domain 3>: <Value 53> Post-Edit: <Post-Edited Sentence> Sentence Pair: English: <English Sentence> Marathi: <Target Language Sentence> Domain Distribution: <Domain 1>: <Value 1>, <Domain 2>: <Value 2>, <Domain 3>: <Value 3> Post-Edit:

Table 18: Prompt used to instruct LLaMA-3.3-70B-Instruct for performing post-editing using a domain distribution.