Giving the Old a Fresh Spin: Quality Estimation-Assisted Constrained Decoding for Automatic Post-Editing

Sourabh Deoghare[1], **Diptesh Kanojia**[2] **and Pushpak Bhattacharyya**[1]

[1]CFILT, Indian Institute of Technology Bombay, Mumbai, India

[2]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) systems often struggle with over-correction, where unnecessary modifications are made to a translation, diverging from the principle of minimal editing. In this paper, we propose a novel technique to mitigate over-correction by incorporating word-level Quality Estimation (QE) information during the decoding process. This method is architecture-agnostic, making it adaptable to any APE system, regardless of the underlying model or training approach. Our experiments on English-German, English-Hindi, and English-Marathi language pairs show the proposed approach yields significant improvements over their corresponding baseline APE systems, with TER gains of 0.65, 1.86, and 1.44 points, respectively. These results underscore the complementary relationship between QE and APE tasks and highlight the effectiveness of integrating QE information to reduce over-correction in APE systems.

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., 2018a). Along with the shift in the field of MT research- from statistical to neural approaches, research within APE has observed a similar trend- towards neural APE systems (Chatterjee et al., 2018a, 2019, 2020).

The need for large APE datasets for training neural APE models is addressed by generating artificial triplets (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018; Freitag et al., 2022). However, unlike real (human post-edited) APE triplets, these do not follow the *minimality principle*, leading to distributional differences (Wei et al., 2020). Despite training on synthetic data and fine-tuning with real data, current APE systems face over-correction issues, primarily due to the size imbalance between synthetic and real data (Chatterjee et al., 2020; Bhattacharyya et al., 2023).

While strategies like optimizing data selection, data augmentation, and model architecture have addressed APE over-correction, mitigating it at the decoding stage remains underexplored (do Carmo et al., 2020). Focusing on other stages limits the applicability across different APE systems. **Motivated** by this, we propose an over-correction mitigation method using an external Quality Estimation (QE) signal during decoding, applicable to any black-box APE system. Our contribution is:

- An over-correction mitigation technique that uses fine-grained word-level QE information to perform constrained decoding. The technique shows improvements of 0.65, 1.86, and 1.44 TER points, respectively, over existing En-De, En-Hi, and En-Mr APE systems (Refer to Table 2).
- Comparison and analysis of the standard beam search and proposed decoding techniques that quantify the extent of how over-correction-prone they are (Refer to Section 5).

2 Related Work

There are multiple attempts to curtail the over-correction at different stages of APE development.

Chatterjee et al. (2016a,b); Wang et al. (2021) focus on data by selecting training samples that may prevent APE from facing the over-correction, augmentation with triplets containing the same translations and post-edits, and weighing training samples with perplexity-based scoring to limit their contribution to learning the APE model.

Junczys-Dowmunt and Grundkiewicz (2017) modify their APE architecture using monotonic hard attention to improve translation faithfulness. Chatterjee et al. (2017) use task-specific loss based on attention scores to reward APE hypothesis

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 914–925

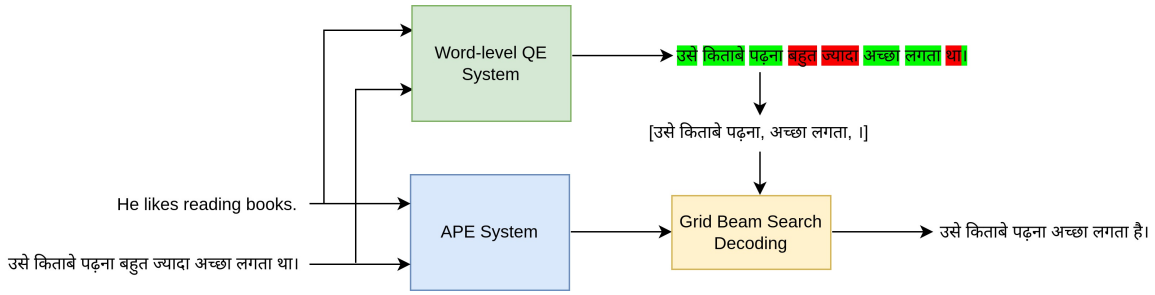

Figure 1: An example of the word-level QE-based Grid Beam Search decoding technique used for English-Hindi APE system. Words marked in green denote word-level QE predicted 'OK' tags for them. These correct translation segments (shown in the list) are referred to as *constraints* and are used during the decoding to ensure they appear in the final APE output.

words present in the original translation. Tebbifakhr et al. (2019) train a classifier to predict post-editing effort and prepend its output to source and translation sequences.

Tan et al. (2017) train separate APE models and use a QE system to rank their outputs. Lee (2020a); Deoghare and Bhattacharyya (2022); Yu et al. (2023) mitigate over-correction by reverting to the original translation based on QE speculation. Chatterjee et al. (2018b) incorporate word-level QE information into the decoder to guide minimal edits. Deoghare et al. (2023b) adopt a multitask approach, jointly training on QE and APE tasks to reduce over-correction. Deguchi et al. (2024) use a detector-correction framework that first predicts the type of edit operation each translation token should undergo, and then the post-edit is generated based on this information.

We find only a few attempts at handling over-correction at the decoding stage. Junczys-Dowmunt and Grundkiewicz (2016) introduce a 'Post-Editing Penalty' during decoding to prevent generating tokens not present in the input, applying it in an ensemble framework to one model. Chatterjee et al. (2017) re-rank APE hypotheses based on precision and recall using shallow features like insertions, deletions, and length ratio, rewarding those closer to the original translation. Lopes et al. (2019) impose a soft penalty for new tokens not in the inputs. Lee et al. (2022) experiment with various decoding methods to generate artificial APE triplets.

3 Methodology

We use an extension of beam search, called Grid Beam Search (Hokamp and Liu, 2017), to perform decoding. While it is originally used for neural interactive-predictive translations and for MT domain adaptation, we adopt the decoding technique for APE. To mitigate the APE over-correction, we explicitly provide information about correct translation segments during the decoding through fine-grained word-level QE signals.

3.1 Grid Beam Search (GBS)

Grid Beam Search (GBS) extends the beam search by incorporating lexical constraints into the sequence generation process. Unlike traditional methods that focus purely on maximizing the probability of the output sequence based on the input, GBS allows specific lexical constraints to be mandatorily included in the generated output.

GBS works by structuring the search space into a grid where the rows track the constraints, and the columns represent the progression of timesteps in the sequence. Each cell in this grid holds a set of potential hypotheses, which are candidate output sequences being considered at that point in time. At each timestep, once a new token is generated, it is matched with the start of tokens in the constraint list. If there is a match, the particular constraint is added to the hypothesis. The algorithm evaluates and updates these hypotheses based on whether they comply with the required constraints and how well they fit the model's learned distribution.

The search proceeds by either continuing with a free generation following the standard beam search or by initiating the enforcement of constraints. This balancing act ensures that, by the end of the sequence generation, all specified constraints are included in the translation. Kindly refer to **Appendix A** for more details.

3.2 Word-QE-based Constraints

A word-level QE system (Ranasinghe et al., 2021) provides fine-grained information about translation

quality by tagging each translation word with an 'OK' or 'BAD' tag. An 'OK' tag indicates the word is a correct translation of some word or phrase in the source sentence. Similarly, a 'BAD' tag denotes the word is an incorrect translation and should be deleted or substituted.

We utilize this information to know the correct translation phrases. We first pass the source sentence and its MT-generated translation to the word-level QE system, which provides tags for each token in the translation. We simply consider a set of consecutive tokens with the 'OK' tag as a constraint that needs to be present in the APE output. Even though the QE system processes the text at the subword level, we set the 'word' to be the smallest unit to be considered as a constraint. Kindly refer to **Appendix B** for details about the word-level QE system.

To summarize, the APE decoding process involves using correct translation segments identified based on the Word-level QE signals and then performing the GBS decoding (Refer Figure 1).

4 Experimental Setup

This section details the different experiments undertaken to assess the effectiveness of the proposed decoding technique. We use the same datasets, architecture, data augmentation, and preprocessing and also follow the same training approach as described by Deoghare et al. (2023b) for training the APE models to enable direct comparison. **Appendix C** details the English-German, English-Hindi, and English-Marathi datasets used for the experiments.

Do Nothing A baseline considering original translations as an APE output.

Baseline 1 (Primary Baseline): Standalone-APE + BS: In this experiment, we train a standalone APE system without any QE data or additionally train the model on QE tasks. The decoding is done using the standard beam search. We consider *Baseline 1* as a **Primary Baseline**.

Baseline 2: QE-APE + BS: The experiment is an extension of *Baseline 1*. In this experiment, the model is jointly trained on QE and APE tasks as described in Deoghare et al. (2023b) by adding QE task-specific heads to the encoders. Similar to *Baseline 1*, this experiment uses the beam search too to perform decoding. This experiment investigates the effectiveness of using word-level QE information during the decoding if the APE model has implicit knowledge of the word-level QE task.

We provide the architecture details and the training approach for both the baselines in **Appendix D** and the hyperparameter information for both APE and QE systems in **Appendix E**.

Standalone-APE + GBS In this experiment, we train the APE model as in the *Baseline 1* experiment. However, the decoding is performed using the proposed Word-QE-based GBS decoding technique.

QE-APE + GBS The experiment involves jointly training a model on QE and APE tasks as in the *Baseline 2* experiment. During decoding, instead of standard beam search, the proposed Word-QE-based GBS decoding technique is used.

Experiment	En-De	En-Hi	En-Mr
Do Nothing	19.06	47.43	22.93
Standalone-APE + BS	18.91	21.48	19.39
Standalone-APE + GBS (Token)	17.40	19.92	18.48
Standalone-APE + GBS (Word)	**17.74**	**19.43**	**17.31**

Table 1: TER scores on the respective evaluation are set in the Oracle settings when constraint enforcement is done based on initial token or word-based matching.

5 Results and Discussion

We perform the experiments on English-German (En-De), English-Hindi (En-Hi), and English-Marathi (En-Mr) pairs, each of which offers a different level of task difficulty due to different linguistic properties, varied amounts of real and synthetic datasets, and 'Do nothing' baselines with different complexities. We use TER (Snover et al., 2006) and BLEU (Papineni et al., 2002) as primary and secondary evaluation metrics, respectively. Kindly refer to Appendix F for the BLEU scores.

Table 1 compiles the results of experiments geared towards answering whether constraint enforcement should be initiated based on the first token match or the entire word match. In *Standalone-APE + GBS (Token)*,' we match the generated token (which is at subword-level, since the 'sentencepiece' tokenization is used) with the first token of each constraint, and if a match is found, the matched constraint is generated. However, in the case of *Standalone-APE + GBS (Word)*,' we wait till the entire word is generated and only then match it with the starting word of each constraint. These experiments are performed in the oracle setting, meaning ground-truth word-level QE tags are used instead of the word-level QE predicted tags to extract correct translation segments. Better perfor-

Experiment	En-De	En-Hi	En-Mr
Do Nothing	19.06	47.43	22.93
Standalone-APE + BS	18.91	21.48	19.39
QE-APE + BS	18.45	19.75	18.30
Standalone-APE + GBS	18.26	19.62	17.95
QE-APE + GBS	**18.04**	**19.20**	**17.53**
Standalone-APE + GBS (Oracle)	17.74	19.43	17.31
QE-APE + GBS (Oracle)	17.50	18.52	16.70
Greedy	19.38	20.04	18.73
Sampling	19.35	19.89	18.46
top-k Sampling	18.43	19.46	18.18
Lopes et al. (2019)	18.38	19.41	18.16
Deguchi et al. (2024)	18.40	19.93	18.92

Table 2: TER scores on the respective evaluation sets in the Oracle and non-oracle settings when different decoding techniques are used. Unlike other techniques, the technique proposed by Deguchi et al. (2024) is not a decoding technique and uses information about edit operations during the training phase.

mance in the case of all three pairs when the constraint enforcement is done based on word-based matching indicates the possibility of noise inclusion, as there could be common subword-level prefixes for multiple words that are present across constraints or even non-constraint words.

A relatively large difference between *Standalone-APE + GBS (Token)* and *Standalone-APE + GBS (Word)* experiments for En-Hi, En-Mr pairs, and En-De pair hints the noise illusion goes up when target languages are morphologically richer. As we observe consistently better results in the case of *Standalone-APE + GBS (Word)* experiment, further experiments are performed by using word-based matching for enforcing constraints during the GBS decoding.

A comparison between different decoding techniques and the proposed technique is depicted in Table 2. We observe larger improvements with the proposed decoding technique *(Standalone-APE + GBS)* over the standard beam search decoding *(Standalone-APE + BS)* when the underlying APE system is a standalone system that is not trained for QE tasks. It shows the effectiveness of enforcing the generation of correct translation segments during the decoding.

On the other hand, a smaller difference in improvements between the two techniques *(QE-APE + GBS vs QE-APE + BS)* when the underlying APE system is jointly trained on QE and APE tasks underlines that the implicit knowledge of the QE tasks helps the model perform APE. Yet, we can conjecture from the better performance with the use of the proposed method over the standard beam search that a loose coupling of QE with APE but with explicit information about the translation segment quality has the potential to improve an APE system developed through the stronger QE and APE coupling.

In both cases, the difference between the proposed technique with oracle and non-oracle word-level QE information underscores the need for better word-level QE systems.

We additionally perform experiments with other popular decoding techniques like greedy, sampling, and top-k sampling for completeness. The *Standalone-APE* model is used in these experiments. The results show that the top-k sampling decoding performs similarly to the beam search decoding. The reported results are with the best k values for each pair (En-De: 25, En-Hi: 30, En-Mr: 25) as per empirical observations.

Comparison with Existing Techniques We also compare our proposed approach with the work of Lopes et al. (2019), who apply a soft penalty during decoding if APE generates tokens that are not present in either source or translation vocabularies. For this experiment too, we use the standalone APE (Standalone-APE) system. While we observe significant improvements in the case of En-Hi and En-Mr pairs, the technique shows limited gains when compared to the proposed approach, suggesting it is more beneficial to inform APE about what to generate than what not to generate since NMT outputs are usually of high quality and require minimal editing.

Furthermore, even though the key aim of this work is to develop an over-correction mitigation technique that could be integrated with any neural network-based APE system, we still compare our proposed technique with existing work that uses the edit operation or QE information at the time of training the APE models. Due to the experimental setup consistency between this work and of Deoghare et al. (2023b), the *Standalone-APE + BS* experiment represents their technique. Its comparison with the *Standalone-APE + GBS* suggests QE-assisted constrained decoding could be more robust in handling the over-correction than relying on the implicit learning of the QE information by the model. Similarly, the comparison with the technique proposed by Deguchi et al. (2024) that relies on the edit operations prediction capabilities of the model shows comparable performance improvements with the performance of our technique.

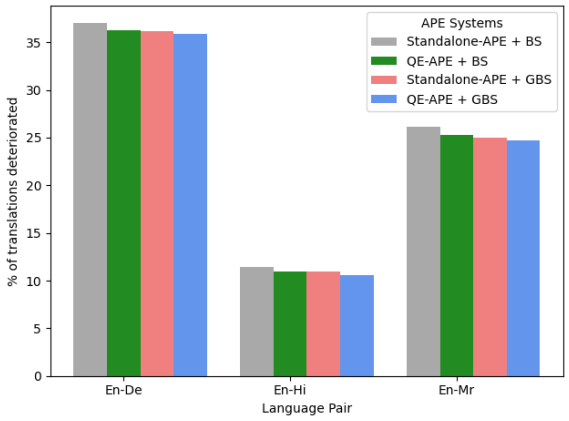

Figure 2: Distribution of percentage of different decoding-based APE model outputs with poorer quality than the original translation.

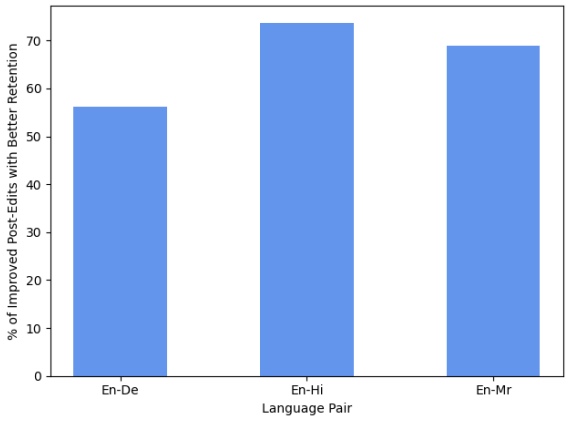

Figure 3: Percentage of post-edits with better retention of correct translation words out of all the improved post-edits from *Standalone-APE + GBS* over the post-edits from *Standalone-APE + BS*.

Deterioration Analysis We analyze the number of translations deteriorated by different decoding techniques to see whether the proposed decoding technique can lead to enforcing undesirable constraints that could lead to poorer post-edit than the original translation. Figure 2 depicts relatively less number of deteriorated translations through the use of our proposed decoding technique over the standard beam search decoding, which points to a reduction in over-correction as the number of APE outputs with poorer quality than the original translation reduces.

Retention Analysis To further assess whether the overall improvement in the TER score is genuinely attributed to a reduction in over-correction, we conduct a retention analysis. Specifically, we compare post-edits from the *Standalone-APE + GBS* experiment with those from the *Standalone-APE + BS* experiment. Our analysis involves computing the percentage of improved post-edits (as determined by TER scores) that contain a higher number of correctly retained translation words. As illustrated in Figure 3, the high percentage of post-edits exhibiting better retention highlights the robustness of the proposed technique in mitigating over-correction.

The statistical significance test (Graham, 2015) considering the primary metric (TER) and p being < 0.05 shows *Standalone-APE + GBS* experiments show significant gains over their *Standalone-APE + BS* counterparts for all three language pairs. Similarly, improvements through *QE-APE + GBS* over *QE-APE + BS* for all three pairs are significant.

6 Conclusion and Future Work

The proposed decoding technique in this work has demonstrated its effectiveness in enhancing the quality of APE outputs by enforcing the generation of provided correct translation segments during decoding. These segments are extracted with the help of a word-level QE system, which offers fine-grained information about translation quality. Through experiments on three language pairs, En-De, En-Hi, and En-Mr, the technique achieved improvements of 0.87 to 2.28 TER points over baseline APE systems. Notably, the superior performance of standalone APE systems using the proposed decoding method compared to QE-APE systems with traditional beam search decoding underscores the technique's ability to reduce over-correction. This result also suggests that injecting word-level QE information exclusively at the decoding stage is more effective than embedding it implicitly through joint QE and APE training. However, the relatively smaller gains when applying the technique to QE-APE systems imply that incorporating explicit QE information at the decoding stage addresses remaining gaps even after joint training with QE and APE.

In the future, we would like to investigate the impact of the quality of a word-level QE system on the proposed decoding technique.

7 Limitations

Our technique relies on the availability of a word-level QE system for the language pair of interest. It limits its applicability to a wider set of languages.

Furthermore, the results show performance improvements through the proposed technique over the standard beam search are sensitive to the quality of the word-level QE system, which is uncontrolled by nature. The false positives of the word-level QE system will especially lead to the enforcement of the decoding technique to include incorrect translation segments in the output.

8 Ethics Statement

Our models for APE and QE are developed using publicly accessible datasets cited in this paper. These datasets have already been gathered and annotated, and this study does not involve any new data collection. Additionally, these datasets serve as standard 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.

References

Farhad Akhbardeh, 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 *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Akanksha Bansal, 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)*.

Pushpak Bhattacharyya, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2022. Findings of the WMT 2022 shared task on automatic post-editing. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 109–117, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Pushpak Bhattacharyya, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2023. Findings of the WMT 2023 shared task on automatic post-editing. In *Proceedings of the Eighth Conference on Machine Translation*, pages 672–681, Singapore. Association for Computational Linguistics.

Ondřej Bojar, 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 *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, Mihael Arcan, Matteo Negri, and Marco Turchi. 2016a. Instance selection for online automatic post-editing in a multi-domain scenario. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 1–15, Austin, TX, USA. The Association for Machine Translation in the Americas.

Rajen Chatterjee, José G. C. de Souza, Matteo Negri, and Marco Turchi. 2016b. The FBK participation in the WMT 2016 automatic post-editing shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 745–750, Berlin, Germany. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: FBK's participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018a. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Frédéric Blain, and Lucia Specia. 2018b. Combining quality

estimation and automatic post-editing to enhance machine translation output. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 26–38, Boston, MA. Association for Machine Translation in the Americas.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Hiroyuki Deguchi, Masaaki Nagata, and Taro Watanabe. 2024. Detector–corrector: Edit-based automatic post editing for human post editing. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 191–206, Sheffield, UK. European Association for Machine Translation (EAMT).

Sourabh Deoghare and Pushpak Bhattacharyya. 2022. IIT Bombay's WMT22 automatic post-editing shared task submission. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 682–688, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sourabh Deoghare, Paramveer Choudhary, Diptesh Kanojia, Tharindu Ranasinghe, Pushpak Bhattacharyya, and Constantin Orăsan. 2023a. A multi-task learning framework for quality estimation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9191–9205, Toronto, Canada. Association for Computational Linguistics.

Sourabh Deoghare, Diptesh Kanojia, Fred Blain, Tharindu Ranasinghe, and Pushpak Bhattacharyya. 2023b. Quality estimation-assisted automatic post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1686–1698, Singapore. Association for Computational Linguistics.

Félix do Carmo, D. Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2020. A review of the state-of-the-art in automatic post-editing. *Machine Translation*, 35:101 – 143.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Yvette Graham. 2015. Improving evaluation of machine translation quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1804–1813, Beijing, China. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 120–129, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Dongjun Lee. 2020a. Cross-lingual transformers for neural automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 772–776, Online. Association for Computational Linguistics.

Dongjun Lee. 2020b. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Wonkee Lee, Baikjin Jung, Jaehun Shin, and Jong-Hyeok Lee. 2022. Reshape: Reverse-edited synthetic hypotheses for automatic post-editing. *IEEE Access*, 10:28274–28282.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Gowtham Ramesh, 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.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2021. An exploratory analysis of multilingual word-level quality estimation with cross-lingual transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 434–440, Online. Association for Computational Linguistics.

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

Yiming Tan, Zhiming Chen, Liu Huang, Lilin Zhang, Maoxi Li, and Mingwen Wang. 2017. Neural post-editing based on quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 655–660, Copenhagen, Denmark. Association for Computational Linguistics.

Amirhossein Tebbifakhr, Matteo Negri, and Marco Turchi. 2019. Effort-aware neural automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 139–144, Florence, Italy. Association for Computational Linguistics.

Chaojun Wang, Christian Hardmeier, and Rico Sennrich. 2021. Exploring the importance of source text in automatic post-editing for context-aware machine translation. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 326–335, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020. HW-TSC's participation in the WMT 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 293–299, Online. Association for Computational Linguistics.

Jiawei Yu, Min Zhang, Zhao Yanqing, Xiaofeng Zhao, Yuang Li, Su Chang, Yinglu Li, Ma Miaomiao, Shimin Tao, and Hao Yang. 2023. HW-TSC's participation in the WMT 2023 automatic post editing shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 926–930, Singapore. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese*

National Conference on Computational Linguistics, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Grid Beam Search Decoding (Hokamp and Liu, 2017)

Algorithm 1 describes the steps followed to perform the GBS. In the grid, beams are indexed by variables t and c. The t variable denotes the timestep of the search, while c indicates the number of constraint tokens that are included in the hypotheses for the current beam. It's important to note that each increment in c corresponds to one constraint token. In this context, constraints form an array of sequences, where individual tokens can be referenced as constraints$_{ij}$, meaning token j in constraint i. The parameter numC in Algorithm 1 signifies the total count of tokens across all constraints. We can categorize the hypotheses in beams as (i) *Open* hypotheses, which can start a constraint generation or generate new tokens based on the distribution over the vocabulary provided by the model. (ii) *Closed* hypotheses, which can only generate tokens for the current constraint.

t each search step, the candidates in the beam at Grid$[t][c]$ can be generated through three distinct methods:

- The open hypotheses from the beam to the left (Grid$[t-1][c]$) can produce continuations based on the model's distribution $p_\theta(y_i \mid x, \{y_0, \ldots, y_{i-1}\})$.
- The open hypotheses from both the beam to the left and the one below (Grid$[t-1][c-1]$) can initiate new constraints.
- The closed hypotheses from the beam to the left and below (Grid$[t-1][c-1]$) can extend existing constraints.

The model described in Algorithm 1 provides an interface that includes three functions: `generate`, `start`, and `continue`, which create new hypotheses in each of the three specified manners. It is important to note that the scoring function does not need to be aware of the constraints' presence, although it can include a feature indicating whether a hypothesis is part of a constraint.

The beams located at the top level of the grid (where $c =$ numConstraints) hold hypotheses that encompass all constraints. When a hypothesis at this top level produces the end-of-sequence (EOS)

Algorithm 1 Grid Beam Search (GBS)

procedure CONSTRAINEDSEARCH(model, input, constraints, maxLen, numC, k)
  startHyp ← model.getStartHyp(input, constraints)
  Grid ← initGrid(maxLen, numC, k) ▷ Initialize beams in grid
  Grid[0][0] = startHyp
  for t = 1 to maxLen **do**
    for c = max(0, (numC + t) - maxLen) to min(t, numC) **do**
      $n, s, g \leftarrow \emptyset$
      for each hyp ∈ Grid[t-1][c] **do**
        if hyp.isOpen() **then**
          $g \leftarrow g \cup$ model.generate(hyp, input, constraints) ▷ Generate new open hypotheses
        end if
      end for
      if c > 0 **then**
        for each hyp ∈ Grid[t-1][c-1] **do**
          if hyp.isOpen() **then**
            $n \leftarrow n \cup$ model.start(hyp, input, constraints) ▷ Start new constrained hypotheses
          else
            $s \leftarrow s \cup$ model.continue(hyp, input, constraints) ▷ Continue unfinished hypotheses
          end if
        end for
      end if
      Grid[t][c] ← k-argmax$_h \in n \cup s \cup g$ model.score(h) ▷ k-best scoring hypotheses stay on the beam
    end for
  end for
  topLevelHyps ← Grid[:][numC] ▷ Get hypotheses in top-level beams
  finishedHyps ← hasEOS(topLevelHyps) ▷ Finished hypotheses have generated the EOS token
  bestHyp ← argmax$_h$ ∈ finishedHyps model.score(h)
  return bestHyp
end procedure

token, it can be included in the collection of completed hypotheses. The hypothesis with the highest score from this set is identified as the optimal sequence that satisfies all constraints.

B Word-level QE System Description

We approach the word-level QE task as a classification problem at the token level. To predict the word-level labels (OK/BAD), we perform a linear transformation followed by a softmax function on each input token derived from the final hidden layer of the XLM-R model.

$$\hat{y}_{word} = \sigma(W_{word}^{T} \cdot h_t + b_{word}) \quad (1)$$

where t indicates the specific token that the model is tasked with labeling within a sequence of length T, $W_{word} \in \mathcal{R}^{D\times 2}$ represents the weight matrix, and $b_{word} \in \mathcal{R}^{1\times 2}$ denotes the bias. The cross-entropy loss function used for training the model is illustrated in Equation 2, which resembles the architecture of MicroTransQuest as detailed by Ranasinghe et al. (2021).

$$\mathcal{L}_{word} = -\sum_{i=1}^{2} \Big(y_{word} \odot \log(\hat{y}_{word}) \Big)[i] \quad (2)$$

Architecture and Training Approach: We utilize a transformer encoder to construct the QE models. For generating representations of the input, which consists of the concatenated source sentence and its translation, we use XLM-R (Conneau et al., 2020). This model has been trained on an extensive multilingual dataset totaling 2.5TB, encompassing 104 different languages, and employs the masked language modeling (MLM) objective, akin to RoBERTa (Zhuang et al., 2021). Notably, the systems that won the WMT20 shared task for sentence- and word-level QE incorporated XLM-R-based models (Ranasinghe et al., 2020; Lee, 2020b). Consequently, we implement a similar approach for our word-level QE tasks. To enable token-level classification for word-level QE, we add a feed-forward layer atop XLM-R. We train these models based on XLM-R for each language pair using their corresponding word-level QE task datasets. Throughout the training process, the weights of all layers in the model are adjusted.

C Datasets

For our experiments, we utilize datasets from the WMT21 (Akhbardeh et al., 2021), WMT24[1], and WMT22 (Bhattacharyya et al., 2022) APE shared tasks for English-German, English-Hindi, and English-Marathi, respectively. The datasets for these language pairs comprise 7K, 18K, and 7K real APE triplets, along with 7M, 2.5M, and 2.5M synthetic APE triplets. However, to facilitate a direct comparison with previous studies (Deoghare et al., 2023a), we limit the English-German pair to 4M synthetic triplets. Each pair also has a corresponding development set containing 1K triplets for evaluation purposes.

In addition, we incorporate parallel corpora during the APE training process. For the English-Hindi and English-Marathi pairs, we draw upon the Anuvaad[2], Samanantar (Ramesh et al., 2022), and ILCI (Bansal et al., 2013) datasets, which each contain approximately 6M sentence pairs. For the English-German pair, we utilize the News-Commentary-v16 dataset from the WMT22 MT task, which consists of around 10M sentence pairs.

For the QE tasks, we also leverage datasets from the WMT21, WMT22, and WMT24 Sentence-level and Word-level QE shared tasks. The English-German QE dataset includes 7K instances for training and 1K for development. The English-Marathi dataset consists of 26K training instances and 1K for development. For English-Hindi, we used the QE-corpus-builder[3] to gather annotations for translations based on their post-edits.

D APE System Description

Architecture: We design the *Standalone-APE* system using a transformer-based encoder-decoder model. For English-Hindi and English-Marathi, two separate encoders are employed to process the source sentence and its translation, as these languages have different scripts and vocabularies. The outputs from both encoders are fed into two sequential cross-attention layers in the decoder. In contrast, the English-German APE system utilizes a single-encoder, single-decoder architecture due to the shared script and vocabulary between these languages. Here, the source and translation are concatenated with a '<SEP>' tag, and this is en-

[1] WMT24 QEAPE Shared Subtask
[2] Anuvaad Parallel Corpus
[3] https://github.com/deep-spin/qe-corpus-builder

coded by a single encoder, which is passed to a cross-attention layer in the decoder. For both language pairs, the encoders are initialized with IndicBERT (Kakwani et al., 2020) weights.

The only change in terms of the architecture for *QE-APE* is the addition of task-specific (Sentence-level QE and Word-level QE) heads on top of a shared representation layer that takes inputs from the last encoder layers. The representation layer has twice as many neurons for the English-Hindi and English-Marathi pairs compared to the English-German pair, whose size matches that of the final encoder layer. While the *Standalone-APE* is trained only for the APE task with cross-entropy loss, the *QE-APE* is trained jointly for sentence-level sentence-level QE (regression), Word-level QE (token-level classification) and APE tasks, with the Nash-MTL (Navon et al., 2022) algorithm used for the optimization.

Data Augmentation and Preprocessing We enhance the synthetic APE data by incorporating automatically generated phrase-level APE triplets. Initially, we train phrase-based statistical machine translation (MT) systems for both source-to-translation and source-to-post-edit tasks using Moses (Koehn et al., 2007). In the subsequent step, we extract phrase pairs from both MT systems. APE triplets are then created by aligning the source sides of the extracted phrase pairs. To ensure the quality of the synthetic APE triplets, including the phrase-level ones, we apply LaBSE-based filtering (Feng et al., 2022) to eliminate low-quality entries from the synthetic APE dataset. This filtering process involves calculating the cosine similarity between the normalized embeddings of a source sentence and its corresponding post-edited translation, retaining only those triplets with a cosine similarity exceeding 0.91. We obtain approximately 45K phrase-level triplets for the English-Hindi pair, around 50K for English-Marathi, and about 60K for the English-German pair.

Training Approach We employ a Curriculum Training Strategy (CTS) for training our APE systems, similar to the approach described by Oh et al. (2021). This strategy involves progressively adapting the model to increasingly complex tasks. The steps of the CTS are outlined as follows.

Initially, we train a single-encoder single-decoder model for translating between the source and target languages using the parallel corpus. Next, we enhance the encoder-decoder model for the English-Hindi and English-Marathi APE systems by adding an additional encoder while maintaining the same architecture for the English-German APE. We train the resulting model for the APE task using synthetic APE data in two phases for English-Hindi and English-Marathi and one phase for English-German. In the first phase, the model is trained using out-of-domain APE triplets. The second phase involves training with in-domain synthetic APE triplets. Finally, we fine-tune the APE model with in-domain real APE data.

Experiment	En-De	En-Hi	En-Mr
Do Nothing	68.79	38.08	64.51
Standalone-APE + BS	68.91	64.79	68.35
QE-APE + BS	69.53	66.56	69.72
Standalone-APE + GBS	69.78	66.52	69.99
QE-APE + GBS	**70.04**	**66.91**	**70.47**
Standalone-APE + GBS (Oracle)	70.37	66.62	70.68
QE-APE + GBS (Oracle)	70.66	67.72	71.31
Greedy	68.42	66.25	69.29
Sampling	68.43	66.43	69.56
top-k Sampling	68.35	66.60	69.84
Lopes et al. (2019)	69.52	66.66	69.89
Deguchi et al. (2024)	69.55	66.41	69.14

Table 3: BLEU scores on the respective evaluation sets in the Oracle and non-oracle settings when different decoding techniques are used. Unlike other techniques, the technique proposed by Deguchi et al. (2024) is not a decoding technique and uses information about edit operations during the training phase.

E Training Details

Our APE models were trained with a batch size of 32 and allowed a maximum of 1000 epochs, incorporating early stopping with a patience of 5. We utilized the Adam optimizer with a learning rate of 5 x 10^{-5}, where β_1 is set to 0.9, and β_2 is set to 0.997. Additionally, we implemented 25,000 warm-up steps. For decoding, we used beam search with a beam size of 5. In the QE experiments, a batch size of 16 was employed, starting with a learning rate of $2e-5$ and using 5% of the training data for warm-up. We also applied early stopping with a patience of 20 steps in the QE and all MTL-based experiments, using WandB for hyperparameter searches. All experiments were conducted on NVIDIA A100 GPUs. The APE model comprises approximately 40 million parameters, with training using the CTS taking around 48 hours, while the QE model contains about 125 million parameters and requires roughly 2.25 hours for training. For preprocessing the English and German datasets, we

used the NLTK library[4], and the IndicNLP library[5] was used for processing Marathi text. Model training and inference were carried out using Pytorch[6]. To compute the TER scores, we utilized the official WMT APE and QE evaluation script[7], and for BLEU scores, we employed the SacreBLEU[8] library.

F BLEU Scores

Table 3 reports BLEU scores for the experiments presented in Table 2.

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

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

[6] https://pytorch.org/

[7] https://github.com/sheffieldnlp/qe-eval-scripts

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