LLMMend: Pinpointing and Refining Large Language Models via Fine-Grained Actionable Feedback

Anonymous ACL submission

Abstract

Recent large language models (LLM) are leveraging human feedback to improve their generation quality. However, human feedback is costly to obtain, especially during inference. In this work, we propose **LLMMend**, an inference time optimization method to refine LLM's output. The core idea is to use a learned fine-grained feedback model to pinpoint defects and guide LLM to refine them iteratively. Using original LLM as a proposal of edits, LLMMend searches for defect-less text via simulated annealing. We conduct experiments on three text generation tasks, including machine translation, long-form question answering (QA), and topical summarization. LLMMend consistently outperforms all baseline approaches, achieving improvements up to 1.7 MetricX points on translation tasks, 8.1 ROUGE-L on ASQA, 2.2 ROUGE-L on topical summarization.

1 Introduction

In recent years, large language models (LLMs) have shown impressive performance on various text generation tasks (Brown et al., 2020; Anil et al., 2023). Critical to their success has been the ability to incorporate human feedback into the learning process (Ouyang et al., 2022).

Nevertheless, human feedback is costly to collect, especially at inference time when the model provides new, unseen input. In the meanwhile, automatic text generation evaluation metrics for a variety of tasks are rapidly improving (Sellam et al., 2020; Xu et al., 2022; Rei et al., 2020; Xu et al., 2023a,b). Can we use one of these metrics to rectify LLM's generation?

In this work, we propose LLMMend, an inference-time optimization method to improve the quality of generated text. Our LLMMend starts with LLM's initial output, then uses a learned error pinpoint model to provide fine-grained feedback about the location and type of defects in the text. We then use a refinement model (same or another LLM) to follow the feedback instruction and generate candidate text. The fine-grained feedback provides more much precise information about what exactly is wrong in the generated text, resulting in higher quality revision.

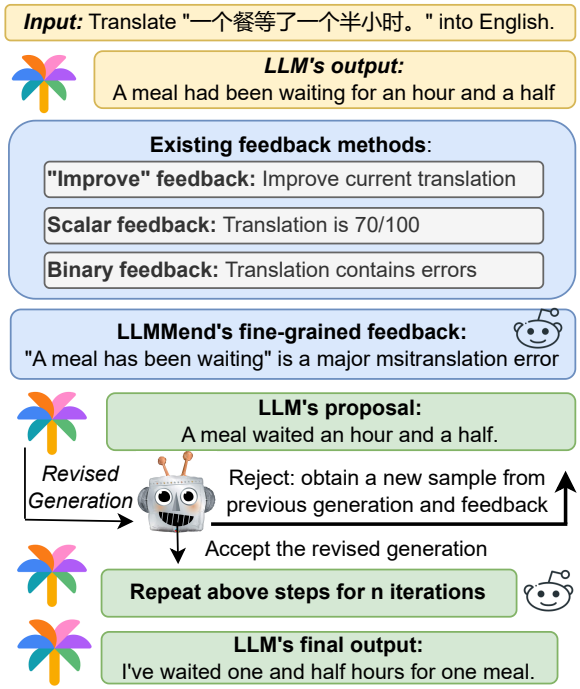

Figure 1: An overview of our LLMMend: We start from LLM's initial generation and iteratively refine the generation, based on fine-grained actionable feedback. We use a simulated annealing technique to accept or reject the proposed revision at each step.

However, due to the large search space, the refinement model is imperfect; it often fails to correct all of the errors identified by the feedback model in one iteration (Madaan et al., 2023). We formulate the iterative refinement procedure into a local search problem. It alternates between the feedback generation and refinement in multiple iterations, with the goal of searching for the highest

scoring output according to the feedback model. To this end, we develop a simulated annealing technique in LLMMend to trade off between exploring many possible edits and quickly reaching optimal text. Figure 1 shows overview of our approach.

We evaluate LLMMend on three text generation tasks, including machine translation (WMT (Kocmi et al., 2022)), long-form question answering (ASQA (Stelmakh et al., 2022)) and topic summarization (Saunders et al., 2022), because they have a large number of annotated outputs with fine-grained error spans (Freitag et al., 2021a; Saunders et al., 2022; Wu et al., 2023). We use those ratings to train an automatic error pinpoint model that generates a list of error spans along with error categories and severities without the aid of a reference text (which is unavailable during inference) (Fernandes et al., 2023; Xu et al., 2023b). This model serves as our substitute for human feedback. Our experiments show that LLMMend results in higher-quality text compared to baseline methods using other feedback (scalar or binary score) or other search techniques. Our contributions are:

- We propose LLMMend, an inference time optimization method to iteratively refine LLM's output with fine-grained actionable feedback, achieving best trade-off between search space and optimal quality.
- We demonstrate that LLMMend consistently outperforms all baseline approaches, achieving improvements up to 1.7 MetricX points on translationn tasks, 8.1 ROUGE-L on ASQA and 2.2 ROUGE-L improvements on topical summarization. Humans demonstrate a significant preference for the output of LLMMend over the baseline outputs.

2 Related Work

Inference-time Optimization Approach We divide techniques for incorporating feedback at inference time into two main techniques (Pan et al., 2023): generate-then-ranking and feedback-guided generation. The reranking framework involves generating a large set of candidate text outputs from the base model and utilizing a critic model to select the best output. The integration of the critic model can be achieved through chain-of-thoughts (Wei et al., 2023; Huang et al., 2022), binary verifier (Li et al., 2023), or a utility function (Freitag et al., 2022a; Fernandes et al., 2022). However, these reranking techniques have limited feedback for candidate selection and lack of control over the decoding process.

To address these limitations, the Plug and Play approach (Dathathri et al., 2020) employs attribute classifiers to guide the pretrained model without additional updates to the model's weights. On the other hand, FUDGE (Yang and Klein, 2021) interpolates token predictions with attribution classifiers at each partial sentence. Building on gradient-based methods, DiffusionLM (Li et al., 2022) performs iterative gradient updates over latent variables to meet specified requirements and extends the pipeline to the diffusion model. In contrast to previous works that primarily focus on scalar or binary feedback from external models, our work considers fine-grained language feedback that identifies the location of errors, the type of errors, and the severity levels associated with them.

Incorporating Fine-Grained Feedback Recent studies have highlighted the benefits of fine-grained error annotation by demonstrating that it can reduce noise in human ratings and increase inter-rater agreement (Freitag et al., 2021b) as well as increase automatic metric correlation to human judgments Xu et al. (2023b). One approach to leveraging these benefits is through the use of large language models to self-correct their own output (Madaan et al., 2023). Building on this, Chen et al. (2023) demonstrate that iterative self-improvement further enhances translation quality. However, despite the unsupervised nature of the self-refine pipeline, the feedback signal is dominated by the large language model's own evaluation capability, which has been shown to be biased towards sentence ordering and its own output (Liu et al., 2023). To address this limitation, Wu et al. (2023) propose a fine-grained reward model that distinguishes rewards at the span-level associating with different error categories. Orthogonal to this work, we propose an inference time optimization approach to iteratively refine model's output with fine-grained language feedback.

3 Refinement with Fine-Grained Feedback

There are three main components to our framework: a generation model, a feedback model, and a refinement model, each described next.

The generation model produces an initial candidate output y_i given the input x. x and y_i are the source text and a candidate output that is generated by the model. The feedback model F takes x and y_i and generates some form of feedback f_i that represents the quality of y_i, which can be in any form—a scalar value, Boolean, free form natural language, or more. We assume f_i can always be converted into a scalar quality score via function $s(\cdot)$ (Details of our scoring scheme can be found in Appendix A). Finally, the refinement model uses x, y_i, and f_i and generates a new, improved output y_{i+1}. As we will discuss in Section 4, the loop between the feedback and refinement model can repeat for multiple iterations to further evaluate and update the generated output.

For most of this work, we assume that both the generation and refinement models are an LLM that is 0-shot prompted to perform the respective task (See example prompt in Table 7, although we do experiment with different generation models). The specific prompt for the refinement model depends on the type of feedback being used (See Figure 1). Since our focus is on the value of fine-grained feedback in the form of an error pinpoint model for text generation, we next describe our feedback model in more detail.

3.1 An Error Pinpoint Model

While the majority of text generation evaluation research focuses on predicting a scalar quality score for a text, we instead train an error pinpoint that produces fine-grained feedback on translation quality, similar to InstructScore (Xu et al., 2023b). This is based on the assumption that more specific, actionable feedback will enable the refinement model to generate better output.

The input to our feedback model is the source text x and a hypothesis generation y_i. The feedback model then generates a list of error locations, types, and severities in natural language that are contained in y_i. We model this task as a sequence-to-sequence model and finetune an LLM. Further implementation details are provided in Section 5.

Training our feedback model requires a set of text with human-annotated error locations, categories, and severities. For each task that we experiment on, the training data and feedback models are different since the types of errors are task-dependent. For machine translation, we use MQM annotated data (Mariana, 2014; Freitag et al., 2021a). For long form QA, we use data collected by Wu et al. (2023). For topical summarization, we use data collected by Saunders et al. (2022).

The finegrained feedback model pinpoints the error location and provides detailed error type information and severity level. This stands in contrast to more traditional evaluation metrics like BLEU, ROUGE or BLEURT that assign scalar scores that represent text generation quality. Note that because the feedback model operates during inference, our feedback model does not use a reference to evaluate the text. The specific input and output examples for our feedback model can be found in the Table 9, 10 and 11.

Once feedback f_i is generated, it is passed to the refinement model via prompting (See Figure 1 for example inputs and outputs to the feedback and refinement model). Specific implementation and evaluation details of our error pinpoint model are described in Section 5.1.

4 Iterative Refinement as Search

Although the refinement model receives the output y_i and feedback f_i, it is not always guaranteed to generate the best new output in a single step. Therefore, we experiment with different methods for iterative refinement in which the feedback and refinement loop is repeated until some stopping condition is met.

Iterative refinement can be viewed as a search procedure that is trying to find the optimal y_i for a given x, where "optimal" is measured by the feedback model. Specifically, we model iterative refinement as a local search algorithm in which every possible output is a state in the search space, and each step of the search algorithm starts at some state represented by y_i and moves to y_{i+1}. The goal is to find the highest scoring state.

We explore three different local search algorithms, described next.

4.1 Local Search Algorithms

Given a current output y_i, the local search algorithms begin by sampling a new candidate output c_i from the refinement model given feedback f_i. Then, each algorithm makes a decision about whether it will accept or reject c_i based on some criteria. If the decision is made to accept c_i, then c_i becomes y_{i+1} and the search loop repeats unless the feedback model detects no errors in y_{i+1}. If c_i is rejected, then y_i becomes y_{i+1} and the algorithm

repeats (i.e., a second candidate is sampled from the refinement model for the same output). Each of the three following algorithms differs in how it decides whether to accept or reject the candidate output.

Always Accept. The "always accept" algorithm (AA) will attempt to explore the search space as much as possible by accepting every c_i no matter what.

Greedy Uphill. The greedy uphill (GREEDY) algorithm will only accept c_i if the score from the feedback model for c_i is better the score for y_i. In this case, we ensure that the output does not get worse according to the feedback model.

Simulated Annealing. The AA and GREEDY algorithms each make different trade-offs. AA will always explore the search space, sometimes at the cost of quality, whereas the GREEDY algorithm may do little in terms of search in order to ensure the output quality does not decrease. Here, we propose a search heuristic based on simulated annealing (SA) (Kirkpatrick et al., 1983) that tries to combine the strengths of the two approaches.

The SA search algorithm uses a temperature hyperparameter T that controls output diversity and the probability that c_i is accepted. The probability of acceptance is defined as the following:

$$p(\text{accept}|x, y_i, c_i, T) = \min(1, e^{\frac{s(F(c_i))-s(F(y_i))}{n*T}}) \quad (1)$$

where n is the maximum number of iterations. i is the ith iteration in the pipeline. $T_{i+1} = max(T_i - c * T_i, 0)$. Temperature decays by a constant proportion c. There are two factors contribute to increasing the probability that a candidate is accepted: a high temperature and an improvement in quality according to the feedback model.

At the beginning of the search algorithm, the temperature is set to a high value, allowing the algorithm to explore the search space more liberally. This allows the SA algorithm to accept a candidate that is potentially worse than the current one, like the AA algorithm. On each iteration, as the temperature drops, output sample becomes more deterministic and this encourages the model to only accept candidates that are better than the current one, like the GREEDY algorithm. In this way, SA combines the strengths of both alternative search procedures. The pseudocode for the SA algorithm can be found in Algorithm 1.

Algorithm 1: Simulated Annealing for Iterative Refinement

Input: Input prompt x, Feedback model F, Base model M
Initialize: $y_0 \leftarrow greedy_decode(M(x))$, T_0, n # Initialize candidate, temperature, constant
for $i = 0..n$ **do**
  $f_i \leftarrow F(x, y_i)$ # generate feedback for the current candidate proposal
  $c_i \leftarrow Sampling(M(x, y_i, f_i))$ # Sample next candidate based on prior one and feedback
  $p_{\text{acc}} \leftarrow \min(1, e^{\frac{s(F(c_i))-s(F(y_i))}{n*T_i}})$
  if *Accept* **then**
    $y_{i+1} \leftarrow c_i$
  else
    $y_{i+1} \leftarrow y_i$
  $T_{i+1} = max(T_i - c * T_i, 0)$ # update temperature for the next iteration
Output: Sampled sequence y_n with n iterations

5 Experimental Setup

Here, we describe the setup for experiments related to implement/evaluate error pinpoint feedback model and implement/evaluate the proposed refinement and iterative refinement procedures.

5.1 Error Pinpoint Model Implementation and Evaluation

We leverage the MQM, ASQA and topic summarization datasets to train and meta-evaluate our error pinpoint model. The model was initialized with PaLM (Bison) LLM and was trained separately for each language pair and each task using WMT'21 MQM data for Zh−En (17,185 examples) (Freitag et al., 2021c), WMT'20 + WMT'21 MQM data for En−De (35,340 examples) (Freitag et al., 2022b), 2853 annotated ASQA examples (Wu et al., 2023) and 17,872 annotated topical summarization examples (Saunders et al., 2022). We use batch size 32, drop out rate 0.1, 500 warm up steps and learning rate $1e-4$ for all languages and tasks. We set maximum prefix length to be 2048 and maximum decoding step to be 512. Checkpoint selection was done by selecting the best Pearson correlation on four separate held-out sets, each comprising 500, 500, 500 and 2000 samples respectively. Each set corresponds to Zh-En translation, En-De translation, long form QA and topical summarization. For the reproducing purpose, we perform greedy decoding for the fine-grained feedback generation. We meta-evaluate the error pinpoint model by calculating Pearson correlation and pairwise accuracy between

its score and ground-truth human ratings. We evaluate pinpointed error span using character-level precision/recall/F_1 scores (Blain et al., 2023). We include details of our meta-evaluation in the Appendix B.

5.2 Refinement Implementation and Evaluation

Datasets & Evaluation Metrics. We evaluate the quality of the text generation produced by our refinement model and search algorithms on the WMT 22 and 23 English-German, Chinese-English (Kocmi et al., 2022), ASQA (Stelmakh et al., 2022) and topic summarization (Saunders et al., 2022) testing set. Each consists of 2037, 557, 1875, 1976, 948 and 284 samples respectively. The translations are automatically evaluated using MetricX (Freitag et al., 2022b), a state-of-the-art reference-based evaluation metric for MT. ASQA and summarization are evaluated by ROUGE-L (Lin, 2004).

Baseline Feedback Models. We experiment with different baseline forms of feedback for the generated output: 1) IMPROVE: The refinement model is always prompted to improve the translation without being provided any feedback. 2) SCORE-QE: The refinement model is provided the score from our error pinpoint model and prompted to improve the output. 3) BINARY-QE: The refinement model is prompted to improve the output given that if our error pinpoint model spots errors. 4) BLEURT-SCORE-QE: The refinement model is provided the score from the BLEURT-QE metric and prompted to improve the output. 5) BLEURT-BINARY-QE: The refinement model is prompted to improve the output given that the BLEURT-QE score for the output is below a hyperparameter threshold. The BLEURT feedback models are only available for the MT task. The prompt used for the refinement model varies based on the type of feedback. See Appendix Table 8 for the specific prompts that were used.

Generation and Refinement Models. The majority of our experimentation uses the PaLM-2 (Bison) LLM (Anil et al., 2023) for both initial translation generation as well as the refinement model. In each case, the LLM was 0-shot prompted for the task. We also experimented with alternative generation models to understand whether the feedback and refinement models could improve their translations, too. In particular, we used translations from the systems submitted to the WMT'22 General Machine Translation Task (Kocmi et al., 2022).

Baseline Generation Model. We compare the outputs from our proposed refinement models to that of 0-shot prompted PaLM-2. This model serves as a comparable baseline and allows us to understand how incorporating feedback via the refinement model can improve the original generation quality.

Implementation Details. The threshold for determining whether or not an error exists for the BLEURT-BINARY-QE model was set to be 0.95 and 1 for Zh-En and En-De, respectively.[1] They were chosen using the held-out WMT'21 test set (Akhbardeh et al., 2021).

For the reproducing purpose, we conduct all single step refinement using greedy decoding. For iterative refinement, we use top-k sampling, with k=40. For the uphill and always accept algorithms, temperature was set to default value 0.8. For the SA search algorithm, the initial temperature was set to 0.8 and is reduced by 10% on each iteration. We experiment different normalization constants from 1 to 10 on our development set WMT21 and choose the best performed constant 4 during simulated annealing. For iterative improvement, we set the maximum number of iterations n to be 10.

6 Results

We explore several research questions in our experiments: 1) How well does our error pinpoint model align with human annotations of translation quality? 2) Does fine-grained feedback result in better downstream translations than more coarse feedback? 3) Can the feedback and refinement models be used to improve translations generated by unrelated text generation models? 4) Does the iterative refinement improve the generated translation quality?

6.1 Meta-Evaluating the Pinpoint Model

Table 1 contains the segment-level meta-evaluation results for our error pinpoint model, BLEURT-QE, and COMET-QE. In all but one evaluation setting, our feedback model has the best results compared to the strong baseline metrics. Therefore,

[1] The BLEURT-QE scores are mostly between 0 and 1, but high-quality translations often receive a score > 1.

Metric	Zh-En		En-De	
	Acc.	r	Acc.	r
COMETKiwi-QE	0.516	0.509	0.583	**0.432**
BLEURT-QE	0.523	0.385	0.591	0.392
Error Pinpoint Model	**0.535**	**0.516**	**0.601**	0.394

Table 1: Segment-level accuracy (after performing tie calibration) and Pearson's r of our error span detection model compared to other reference-free evaluation metrics on the WMT'22 zh-en and en-de datasets.

Metric	En-De			Zh-En		
	P	R	F_1	P	R	F_1
AutoMQM (Bison)	0.05	0.58	0.09	0.10	0.17	0.13
XCOMET-XXL	0.24	0.38	0.29	0.15	0.57	0.24
Error Pinpoint Model	**0.28**	0.21	0.24	**0.30**	0.31	**0.30**

Table 2: Character-level precision/recall/F_1 of different reference-free error span tagging models. Our error pinpoint model has the highest precision compared to others. This implies that our predicted error spans are most reliable.

we conclude that the feedback model is a state-of-the-art evaluation metric and is a high-enough quality to be used in the rest of our experiments.

Table 2 contains the automatic evaluation of the predicted spans. Among the metrics, our feedback model achieves the highest Character-level precision on both language pairs and the best Character-level F_1 on Chinese-English, making it a suitable candidate for identifying errors that should be corrected during the refinement step of our pipeline. In Appendix C, we meta-evaluate our error pinpoint model by comparing the gap in downstream translation quality when human-annotated error spans are used. You can find input output examples of our error pinpoint model for each task at Appendix Table 9,10 and 11

6.2 Fine- vs. Coarse-grained Feedback

Table 3 compares the quality of the refined translations when different forms of a feedback are used plus the PaLM-2 0-shot baseline quality.

Inadequate feedback could deteriorate the generation. While always prompting the refinement model to improve (IMPROVE) exhibits better translation performance for WMT'22 and WMT'23 when focusing on Zh−En, it results in a significant decline in translation quality for En-De. This highlights the instability of the direct prompting approach. Similar patterns are observed when examining using only scalar feedback scores from BLEURT-SCORE-QE and only refining translations when the metric predicts there is an error (BLEURT-BINARY-QE). We observe steady performance improvements by adding more detailed feedback at translation, long form question answering and Topical summarization. We argue that the lack of detailed error analysis increases the task difficulty and can't fully elicit LLM's refinement ability.

By contrast, fine-grained feedback from our error pinpoint model delivers significant and consistent improvements: Using our fine-grained feedback model with a single iteration consistently enhances the quality of the base translation in both Zh−En and En−De across all four testing sets and achieves the highest performance at ASQA and topic summarization. In Appendix D, we show that granularity of fine-grained feedback (e.g. inclusion of severity or error type) can further impact the model refinement.

Examining Generations with Errors Many of the original outputs from our PaLM-2 generation model are already error free according to our error pinpoint model. In such cases, no refinement is done and the result does not change, so the magnitude of the MetricX or ROUGE-L improvement made by the refinement model is not well represented. Therefore, we additionally report results on the subset of the WMT'22, '23, ASQA and topical summarization datasets for which our feedback model detected an error [2].

From Table 4, we observe the improvements are much larger than on the entire dataset as a whole. For example, on WMT'22 zh-en, the improvement using our error pinpoint model is 2.7 MetricX points compared to 0.6 on the full dataset. This further demonstrates the effectiveness of our method. When the feedback model detects an error, the refinement model can make significant quality improvements.

6.3 Improving Other Source of Generation

We study the possibility of improving initial translations that come from systems other than PaLM-2, or even improving human translations. We conduct experiments on Zh-En and En-De for 14 submission systems and one set of human translations from WMT22. We performed one step refinement based on fine-grained feedback.

[2]This consists of 407/1875, 329/1976, 465/2037, 334/557, 937/948 and 166/284 on the WMT'22 Zh-En, WMT'23 Zh-En, WMT'22 En-De, WMT'23 En-De, ASQA and topical summarization respectively

	MT22 Zh−En	MT23 Zh−En	MT22 En−De	MT23 En−De	ASQA	Topical Summ
Baseline	Metric-X	Metric-X	Metric-X	Metric-X	ROUGE-L	ROUGE-L
PALM-2 0-SHOT	75.3	73.8	83.1	78.3	17.6	28.7
Feedback Models						
IMPROVE	75.6	74.0	78.9	77.8	19.2	28.8
BLEURT-SCORE-QE	75.6	74.1	80.0	77.8	-	-
BLEURT-BINARY-QE	**75.9**	74.1	82.3	78.9	-	-
SCORE-QE	75.6	74.0	83.2	79.0	21.9	29.4
BINARY-QE	75.7	74.0	83.3	79.1	21.6	29.1
LLMMEND	**75.9**	**74.2**	**83.5**	**79.3**	**26.1**	**30.5**

Table 3: We include three baseline models using coarse feedback: IMPROVE, BINARY-QE, SCORE-QE, BLEURT-BINARY-QE, BLEURT-SCORE-QE and LLMMend, which is guided by our fine-grained error pinpoint model. All results are obtained through greedy decoding.

Model	WMT'22		WMT'23		ASQA	Summ
	Zh-En	En-De	Zh-En	En-De	QA	Summ
PaLM-2 0-shot	66.1	77.0	65.7	75.1	17.6	25.2
IMPROVE	67.7	77.1	67.5	75.9	19.2	25.5
SCORE-QE	67.5	77.2	67.2	76.3	21.9	26.4
BINARY-QE	67.6	77.9	67.3	76.5	21.6	26.0
LLMMEND	**68.8**	**78.6**	**68.2**	**76.9**	**26.1**	**28.1**

Table 4: Fine-grained feedback vs coarse feedback on the examples that are marked as "errors" by our error pinpoint model. MetricX is used for all translation results and ROUGE-L is used for ASQA and summ.

Our refinement pipeline consistently improves all of the WMT22 systems, with an average improvement of 2.2 MetricX in En−De and 2.8 MetricX in Zh−En (Check Appendix for Figure 5). Notably, it is effective in improving the translation quality of systems that already demonstrated better performance than the PaLM 2 zero-shot translation.

Although the human translations are high-quality, they still contain errors as marked by MQM raters (Freitag et al., 2022b), therefore, there is room for improvement. Indeed, we find that our single-step refinement manages to improve even those by as much as 0.8 MetricX in the Zh-En task, and 0.7 MetricX in En-De.

6.4 Iterative Refinement

Figure 2 contains the results from running the various proposed refinement algorithms for up to 10 iterations. We demonstrate that Always Accept and Greedy Uphill each outperforms another in different test sets due to a trade-off between search space and error feedback. Overall, Simulated Annealing performs best in multi-step refinements. The figure shows that all three algorithms can result in further performance improvements on top of the initial output. Notably, we observe that always accepting the output (AA) demonstrates rapid convergence to the maximum, typically requiring only around 1 or 2 iterations. However, it can demonstrate instability of performance (See the fluctuation of the performance in the right figure) as they are lack of a quality selection process. In contrast, the uphill and simulated annealing techniques yield additional performance improvements over more iterations. We also include detailed iterative results with all tasks for first and fifth iteration at Appendix Table D.

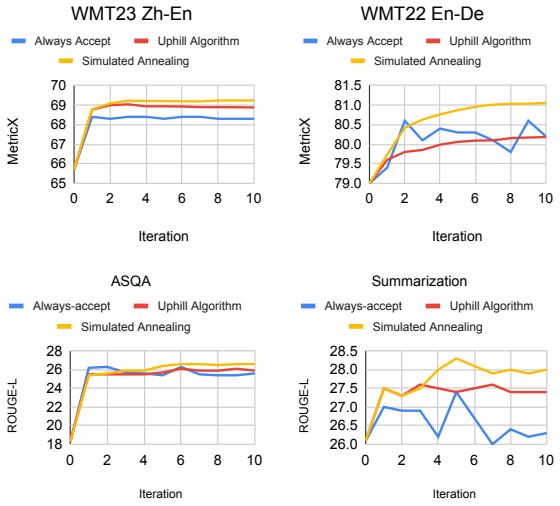

Figure 2: We conducted iterative experiments on WMT23 Zh-En and WMT22 for En-De, ASQA and topical summarization with always accept, greedy uphill and simulated annealing algorithms and report MetricX and ROUGE-L score.

Simulated Annealing finds better candidates. In Figure 2 and 3, we present a comparison of three search algorithms on WMT22 En−De and WMT23 Zh−En, focusing on their performance in terms of MetricX/ROUGE-L and correction rate improvements (identified by our error pinpoint model). Our observations indicate that during the first iteration, simulated annealing may identify

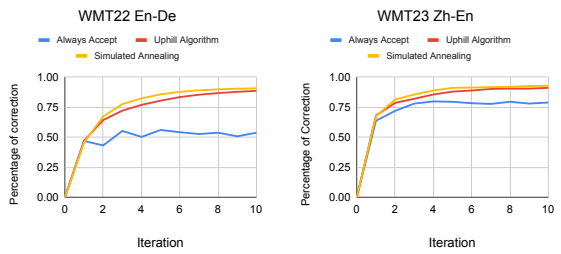

Figure 3: We conducted iterative experiments on WMT22 En-De and WMT23 Zh-En with Always Accept, Uphill and Simulated annealing algorithms and report correction rate of error pinpoint.

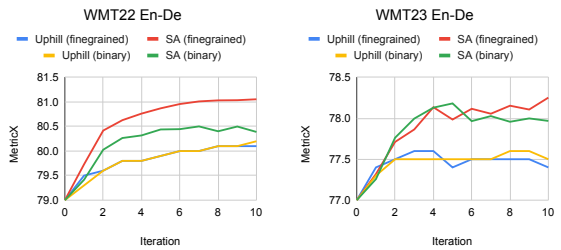

Figure 4: We study whether finegrained feedback can find better candidate generation compared to binary feedback. We conducted experiment on WMT22 and WMT23 at En-De with uphill and simulated annealing and report MetricX.

samples that achieve lower MetricX scores and corrects a smaller number of examples compared to uphill algorithm. However, owing to the diverse candidate proposals generated by SA under the measure of performance improvements, by the second to forth iteration, it already identifies samples that achieve higher MetricX scores. The performance gap between the three algorithms widens over the subsequent four or five iterations, ultimately resulting in a superior candidate proposal at the end of the iterations. We provide a concrete case study example in Appendix Table 6. In contrast to always accept, although with full search space, it could occasionally optimize MetricX on WMT22 En−De or ASQA. It can not fully optimize error correction rate without a quality selection process. On average, simulated annealing improves the uphill algorithm by 0.5 MetricX on Zh−En, 0.9 on En−De and by 0.7 ROUGE-L on ASQA and 0.9 ROUGE-L on topical summarization.

In Figure 4, we empirically show that simulated annealing can boost the performance for different types of feedback (binary and fine-grained). Specifically, we show that simulated annealing with both feedback can significantly improve the proposal quality for their counterparts under uphill algorithm. Furthermore, we demonstrate that simulated annealing with fine-grained feedback can achieve the best MetricX score in additional iterations.

	0-shot	Improve	B-score	B-binary	score-qe	binary-qe
LLMMend	2.34	2.44	2.79	1.76	1.23	1.84

Table 5: We conduct head-to-head human evaluation on LLMMend against all baselines for single step refinement at WMT22 En-De. B-score and B-binary represent BLEURT-SCORE-QE and BLEURT-BINARY-QE respectively. We report win/lose ratio in the table.

6.5 Human Evaluation Results

We conduct head-to-head human evaluation on WMT22 En-De with the samples that marked as errors by our feedback model (465/2037). Professional bilingual annotators were shown the source sentence and asked to rate the quality of test translation against base translation with either win, lose or neutral options. We use the win/lose ratio of test translation against base translation as the final metric. If win/lose ratio is greater than 1, then test translation outperforms base translation. In Table 5, we compared single step fine-grained refinement against all coarse feedback baselines. We found that fine-grained feedback outperforms all other source of feedback, as win/lose are all above 1. Moreover, we compare simulated annealing (SA) baseline against always-accept (AA) and greedy baseline at 5th iteration. We find win/lose ratios are 1.56 and 1.38 respectively, indicating SA has superior performance against AA and greedy. We include the full results of percentages of win, neutral and lose in Appendix Table 14 and 15.

7 Conclusion

In this work, we proposed LLMMend for incorporating a fine-grained error pinpoint into a text generation pipeline via refinement-feedback model loop. We empirically demonstrated that our fine-grained error pinpoint model with simulated annealing achieves superior performance compared to baseline feedback models and other search algorithms across three text generation tasks. Lastly, our experimental findings are further solidified by the human evaluation study. Humans demonstrate a significant preference for the output of LLMMend over the baseline outputs.

8 Limitations

In this work, we demonstrate that our error pinpoint can achieve comparable Pearson correlation and segment-level accuracy to source based SOTA metrics in Machine translation at WMT22 En-De and Zh-En. Despite the great performance, we also demonstrate the trade-off between precision and recall at Table 2. Although achieving higher precision score compared to baseline metrics, our error pinpoint has lower recall. Future work can consider a better pipeline to boost the recall of our error pinpoint while maintaining a reasonably high precision score. Ultimately, this may significantly improve the quality of corrections after iterative refinement.

While LLMMend in theory can be applied to landscape of instruction-fine-tuned large language models, it would be noteworthy to mention that the efficiency may differ when applied with models that lack instruction following capability. Training a large language model with instruction, feedback following ability can be great a future direction to mitigate this issue.

9 Ethical Statement

All the training data of our error pinpoint model is publicly available. We ascertain that the feedback data that is annotated by human labors do not contain risk or toxic content. We used an internal, proprietary tool to collect human evaluation data. The annotators were compensated fairly and did not have to disclose any personal information during the annotation process. All of the test sets used in this study are publicly available, and annotators were allowed to label sensitive information if necessary. The annotators are fully aware that the data which is collected from them will be used for research purposes. Each pair of translations is annotated by one rater. There are six annotators participated for each test vs base system comparisons.

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.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Frederic Blain, Chrysoula Zerva, Ricardo Ribeiro, Nuno M. Guerreiro, Diptesh Kanojia, José G. C. de Souza, Beatriz Silva, Tânia Vaz, Yan Jingxuan, Fatemeh Azadi, Constantin Orasan, and André Martins. 2023. Findings of the wmt 2023 shared task on quality estimation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 629–653, Singapore. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder,

Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023. Iterative translation refinement with large language models.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties Matter: Modifying Kendall's Tau for Modern Metric Meta-Evaluation.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021b. Experts, errors, and context: A large-scale study of human evaluation for machine translation.

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

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022b. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021c. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xCOMET: Transparent Machine Translation Evaluation through Fine-grained Error Detection.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*, 220(4598):671–680.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-lm improves controllable text generation. *ArXiv*, abs/2205.14217.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Valerie Ruth Mariana. 2014. The multidimensional quality metric (mqm) framework: A new framework for translation quality assessment.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies.

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

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte M. Alves, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*.

Wenda Xu, Xian Qian, Mingxuan Wang, Lei Li, and William Yang Wang. 2023a. SESCORE2: Learning text generation evaluation via synthesizing realistic mistakes. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5166–5183, Toronto, Canada. Association for Computational Linguistics.

Wenda Xu, Yilin Tuan, Yujie Lu, Michael Saxon, Lei Li, and William Yang Wang. 2022. Not all errors are equal: Learning text generation metrics using stratified error synthesis.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023b. Instructscore: Towards explainable text generation evaluation with automatic feedback.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

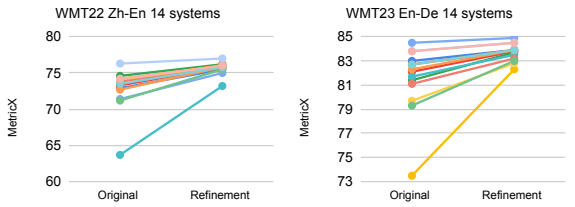

Figure 5: MetricX score improvement after one-step refinement of external systems' translations, evaluated on WMT22 Zh-En and En-De.

A Feedback Scoring Scheme

We adopted the same setting as human evaluation respect to each task. For machine translation and long form question answering, we adopt MQM human scoring scheme (Freitag et al., 2021a) by counting the total number of errors and assigning the weight penalties to the individual error. We use the same severity level (major and minor) and weight penalty as MQM, assigning major error as penalty 5 and minor error as penalty 1. For the sentence without errors, we assign a score of 0. Similar to MQM scheme (Freitag et al., 2021a), for sentence contains more than five major errors, they all assigned to -25. The score range is between 0 to -25. For topical summarization, we adopt the same severity levels as Saunders et al. (2022)'s annotations. In particular, they contain four levels of severity: critical, major, medium and minor. Each response will only be assigned with one error type. Therefore, we score each sentence with 1 to 5 rating, with 5 corresponds to no-error and 1 corresponds to one critical error. We use those scores and human ratings to compute Pearson correlation and pairwise accuracy for meta-evaluating the feedback model. To ensure fair comparisons at our SCORE-QE baseline, we normalize different score ranges into 0 to 100. We include scoring examples of error pinpoint model at Table 9, 10 and 11.

B Meta Evaluation of Error Pinpoint Model

We meta-evaluate the error pinpoint model by calculating the correlation between its scores and ground-truth human MQM scores on the WMT'22 English-German and Chinese-English datasets, two benchmark datasets for meta-evaluating metrics. We calculate a segment score by summing the scores corresponding to the MQM error severity weights that are predicted by our error pinpoint model: 5 for major errors and 1 for minor errors. We report Pearson and pairwise accuracy with tie calibration (Deutsch et al., 2023) at the segment-level and compare to two state-of-the-art reference-free evaluation metrics, COMETKiwi-QE (Rei et al., 2022) and BLEURT-QE, a reference-free version of BLEURT (Sellam et al., 2020) that we trained ourselves on the same data used by COMETKiwi-QE.

To evaluate the actual spans produced by our feedback model, we adopt the character-level precision/recall/F_1 that was used by the WMT'23 QE Shared Task.[3] The evaluation treats each translation character as a classification problem and micro-averages the results. We compare our model to AutoMQM (Fernandes et al., 2023) and XCOMET (Guerreiro et al., 2023) on the news subset of the WMT'22 MQM dataset because XCOMET uses the remainder of WMT'22 for training (the results on the full dataset are available in Appendix E.

C The performance compared to oracle feedback

Next, we perform a similar study, but replacing our feedback model with oracle human feedback. This is to measure the effectiveness of our feedback model in guiding the refinement. Revisiting Figure 5, we can compare the performance improvements achieved in this way (i.e., with a professional annotator's guidance) to those achieved with our feedback model's guidance (see Human vs. Inst-QE). What we find is that the performance of refinement with the feedback model is competitive, with the scores on average a mere 0.2 and 0.3 BLEURT behind those achieved with oracle human feedback for En-De and Zh-En, respectively. This discovery validates the effectiveness of our automatic feedback in improving the quality of the base translation.

D Granularity of the fine-grained feedback impacts the refinement

To further analyze the granularity of fine-grained feedback, we conducted an ablation study on 10647 system outputs for WMT22 Zh−En and 6441 system outputs for WMT22 En−De, all of which were flagged as containing errors by our feedback model. Specifically, we examined the additive effects of each component (error

[3] https://wmt-qe-task.github.io/

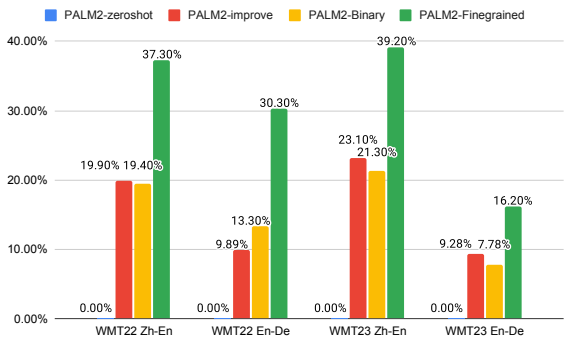

Figure 6: Percentage of the corrections based on feedback model

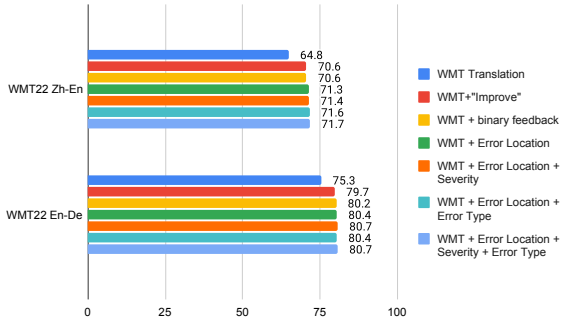

Figure 7: Granularity of the finegrained feedback impacts the refinement performance

location, severity, error type) and their contribution to the overall performance. We observed that providing prompt with error location information significantly improved performance for WMT22 Zh−En, resulting in a BLEURT improvement of 0.7 compared to only mentioning sentences containing errors. Additionally, severity labels and error types each had their own unique additive effects on the final performance. Finally, when all fine-grained feedback, including error type, location, and severity label, were combined, the joint feedback approach achieved the highest performance improvements.

E Additional Results

Table 13 contains the character-level evaluation of the span tagging models, AutoMQM and our error detection feedback model, on the full WMT'22 dataset. XCOMET is omitted because the non-news sections of the dataset were used for training and is thus not a fair comparison.

Source: 一个餐等了一个半小时。 **Reference:** I've waited one and a half hours for one meal. **Output:** A meal had been waiting for an hour and a half.
1st candidate generation: A meal had been waiting for an hour and a half. **Feedback:** Error type: mistranslation Major/minor: major Error location: A meal had been waiting. **Our Error Pinpoint Model:** -5 **BLEURT:** 0.738
2nd candidate generation: "A meal had been waiting for an hour and a half." is a major mistranslation error.The correct translation should be: "The meal had been waiting an hour and a half." **Feedback:** Error type: mistranslation Major/minor: major Error location: A meal had been waiting for an hour and a half. **Our Error Pinpoint Model:** -5 **BLEURT:** 0.692
3rd candidate generation: The meal had been waiting for an hour and a half **Feedback:** Error type: mistranslation Major/minor: major Error location: The meal had been waiting **Our Error Pinpoint Model:** -5 **BLEURT:** 0.781
4th candidate generation: A meal waited an hour and a half. **Feedback:** Error type: mistranslation Major/minor: major Error location: A meal waited **Our Error Pinpoint Model:** -5 **BLEURT:** 0.703
5th candidate generation: A meal took an hour and a half to arrive. **Feedback:** Error type: no-error Major/minor: no-error Error location: None **Our Error Pinpoint Model:** 0 **BLEURT:** 0.876

Table 6: For uphill algorithm, pipeline is stuck at generating "A meal had been waiting for an hour and a half." with BLEURT score 0.738 for five iterations. For the simulated annealing, despite the model's ability at generating candidates with slightly worse BLEURT score. The pipeline yield global optimum solution at the fifth iteration.

Prompt for Translation: Translate "一个餐等了一个半小时。" into English. Your translation:

Table 7: We prompt to obtain initial translation from PALM2.

Prompt for IMPROVE: Translate "一个餐等了一个半小时。" into English. Your translation is "A meal had been waiting for an hour and a half.". Please improve your translation. New translation:
Prompt for SCORE-QE: Translate "一个餐等了一个半小时。" into English. Your translation is "A meal had been waiting for an hour and a half.". Translation quality is 80 out of 100. Please improve your translation. New translation:
Prompt for BINARY-QE: Translate "一个餐等了一个半小时。" into English. Your translation is "A meal had been waiting for an hour and a half.". Your translation contains errors. Please improve your translation. New translation:
Prompt for ERROR PINPOINT: Translate "一个餐等了一个半小时。" into English. Your translation is "A meal had been waiting for an hour and a half.". "A meal had been waiting" is a major mistranslation error. Please improve your translation. New translation:

Table 8: We include refinement prompts for four different forms of feedback: IMPROVE, SCORE-QE, BINARY-QE and ERROR DETECTION.

Prompt for error pinpoint model: Source translation (Chinese): 一个餐等了一个半小时。Candidate translation (English): A meal had been waiting for an hour and a half. You are evaluating Chinese-to-English Translation based on source and candidate translations. Your evaluation will contain error type, location and major/minor labels. **Output for error pinpoint model:** 'A meal had been waiting' is a major mistranslation error.

Table 9: An machine translation example prompt and output we used for our error pinpoint trained from from PALM2. According to our scoring scheme, one major error corresponds to -5 weight penalty. The score is -5 and we normalize it to 80 out of 100.

Prompt for error pinpoint model:
"You are evaluating answer based on the passage. Passage: Drag Me to Hell Her boss advises her to demonstrate that she can make tough decisions. An elderly woman, Sylvia Ganush, asks for a third extension on her mortgage payment, and despite Ganushs financial and medical problems, Christine denies her an extension to prove herself. Ganush begs Christine not to repossess her house. Ganush is taken away, accusing Christine of shaming her and swears revenge. In the parking garage Christine is ambushed by Ganush, who is hiding in the back seat. Ganush rips a button from Christines coat and intones words in another language. Later, Christine and her boyfriend Clay meet fortune teller Rham Jas, who tells Christine that she is being haunted by a dark spirit, likely the result of a curse. At home, Christine is attacked by the entity and has nightmares about Ganush. At work the next day, she hallucinates and bleeds profusely from her nose. She leaves the office, and, amid the general panic, Stu steals a file from Christines desk. Christine goes to beg Ganush for forgiveness but discovers that Ganush has died. Christine returns to Jas, who explains that as long as Christine is the owner of an accursed object (the button), she will be haunted by a powerful demon called the Lamia. Drag Me to Hell Drag Me to HellDrag Me to Hell is a 2009 American supernatural horror film co-written and directed by Sam Raimi. The plot, written with his older brother Ivan, focuses on a loan officer, who, because she has to prove to her boss that she can make the hard decisions, chooses not to extend an elderly womans mortgage. In retaliation, the woman places a curse on the loan officer that, after three days of escalating torment, will plunge her into the depths of Hell to burn for eternity. Raimi wrote Drag Me to Hell with his brother, Ivan, before working on the Spider-Man trilogy. The film premiered at the Cannes Film Festival and was released to critical acclaim. It was also a box office success, grossing over $90 million worldwide. Drag Me to Hell won the award for Best Horror Film at the 2009 Scream Awards and the 2010 Saturn Awards. In 1969, in Pasadena, a couple seeks the aid of the medium Shaun San Dena, saying their son has been hearing evil spirits voices after stealing a silver necklace from a gypsy wagon. San Dena aids the family by carrying out a séance, but they are attacked by an unseen force that drags the boy to Hell. In present-day Los Angeles, bank loan officer Christine is in competition for a promotion with Stu Rubin. Pasadena, California Pasadena, CaliforniaPasadena is a city in Los Angeles County, California, United States, located 10 miles (16 kilometers) northeast of Downtown Los Angeles. The estimated population of Pasadena was 142,647, in 2017, making it the 183rd-largest city in the United States. Pasadena is the ninth-largest city in Los Angeles County. Pasadena was incorporated on June 19, 1886, becoming one of the first cities to be incorporated in what is now Los Angeles County, following the city of Los Angeles (April 4, 1850). It is one of the primary cultural centers of the San Gabriel Valley. The city is known for hosting the annual Rose Bowl football game and Tournament of Roses Parade. In addition, Pasadena is also home to many scientific and cultural institutions, including Caltech, Pasadena City College, Fuller Theological Seminary, ArtCenter College of Design, the Pasadena Playhouse, the Ambassador Auditorium, the Norton Simon Museum, and the USC Pacific Asia Museum. The original inhabitants of Pasadena (a Chippewa word meaning Crown of the Valley) and surrounding areas were members of the Native American Hahamog-na tribe, a branch of the Tongva Nation. They spoke the Tongva language (part of the Uto-Aztecan languages group) and had lived in the Los Angeles Basin for thousands of years. Answer: Drag Me to Hell is a 2009 American supernatural horror film directed by Sam Raimi. The plot focuses on a loan officer who, because she has to prove to her boss that she can make the hard decisions, chooses not to extend an elderly womans mortgage. In retaliation, the woman places a curse on the loan officer that, after three days of escalating torment, will plunge her into the depths of Hell to burn for eternity. The film is set in Pasadena, California and takes place in Pasadenadena, California. Your feedback:"
Output for error pinpoint model:
Answer contains a Missing-Major-Auxiliary error, which misses Drag Me to Hell won the award for Best Horror Film at the 2009 Scream Awards and the 2010 Saturn Awards. from passage
Answer contains a Missing-Answer error, which misses San Dena aids the family by carrying out a séance, but they are attacked by an unseen force that drags the boy to Hell. from passage
directed by Sam Raimi. in the answer is a Irrelevant error. The plot focuses on a loan officer who, because she has to prove to her boss that she can make the hard decisions, chooses not to extend an elderly womans mortgage. in the answer is a Irrelevant error.
In retaliation, the woman places a curse on the loan officer that, after three days of escalating torment, will plunge her into the depths of Hell to burn for eternity. in the answer is a Irrelevant error.
and takes place in Pasadenadena, California. in the answer is a Redundant error.

Table 10: A long form QA prompt and output we used for our error pinpoint trained from from PALM2. According to our scoring scheme, one major error corresponds to penalty of -5 and one minor error corresponds to penalty of -1. The total score is $(-1) * 4 + (-5) * 1 = -9$. The normalized score is 64 out of 100.

Prompt for error pinpoint model:
"You are evaluating a summarization based on question and passage. Passage: It was a cold, dark night... I lay in the corner of the street, my head in a puddle, a smell of what can only be described as death circling my very presence. I had lost count of how many days it had been, of how many faces I've seen pass me, of how many feet I've watched shuffle aside, trying their hardest to avoid my very existence. Of how many eyes had looked at me, and burnt hatred so deep it became almost intolerable. Sheer disgust, seeping out of the breath of everyone who passed. That is what my life had become. Ever since that one day. When everything I thought I knew was ripped apart in front of me.It was the youngest who took me in. He seemed kind, welcomed me. One of those people that deep down you can tell had a good heart even if they hid it beneath a rough, silent exterior. I lived in his room free to do what I felt, unless the others were around, the older ones. When their voices carried through the door, I was hidden under the bed, its so our friendship will be ours alone I would convince myself, that he wanted me to be all his. But I suppose that should have been the first warning sign. Can a place truly be called your home if you have to hide from its very inhabitants? But sadly, I was naive... no, I chose to ignore it, I was too obsessed, too caught up by this newfound friendship to ever even consider the truth. I thought everything was perfect. Then it happened. It was late at night, the door slammed behind him as the boy tumbled in. Raised voices instantly burnt through the very walls that surrounded me, through the doors, through everything. I had heard anger in those voices before, the times when I hid, but nothing like this, this was... new. The door crashed open, the boy collapsing to the ground. Eyes bloodshot, his mouth lined with dry, cracked vomit, his shirt, blooded, torn and stained. A sight that I wished I would never see again. As he hit the ground, he looked up at me, but there was no affection any more, just pure emptiness. I heard the voices come closer. There was no time to hide, no time to be hidden. Voices entered the room; eyes were cast down the decrepit shell lying on the floor, then to me. Hatred. Rough hands were placed on me, words crying out that I was to blame, that somehow, this was my entire fault. My protests, my cries of defense, all fell on deaf ears. No matter how much I pleaded, no matter how many times I cried out to them that there was more to me, more I could give to them, it was all to no avail. I was tossed out on the street, my once home fading in my eyesight as I tumbled, seemingly forever, sinking in to my own personal abyss. An icy wind blew threw me, bringing me back from the horrid memory. I rolled onto my side, no longer sure if it was by my own personal doing or if I simply lacked the strength to offer any resistance. This is it, I thought to myself. This is how it is going to end. All the things I could have done, the places I could have seen, and now this will become my final resting place, a blotch in the street, surrounded by the ear wrecking sound of traffic and drunks. Some rest indeed. Through all the noise, I heard footsteps walking towards me. No doubt to impose some form of abuse towards me in my final moments I had thought. But no, a hand rested on my body. Warmth that I had not felt since... the boy? I turned around and stared up, locking eyes with an unfamiliar face. It was irrelevant. This man has picked me up in my time of need. He had saved me. took me to some form of haven for my kind. After I adjusted to the contrast of light, I looked around and saw countless amounts of those in the same position as me. It was amazing. I was no longer a freak. No longer an outcast, I was amongst my own kind at last. The time I spent there was the happiest of my life. On occasions, members of our private little community would be taken away by strangers from the outside, I had lost a few good friends through that, but even then the kind man would always bring in new members, new friends to be made, stories to be heard, connections to be formed. I should have learnt. All good things must come to a pass. The horrendous noise sounded, the noise that sounded the arrival of a stranger. As usual, our host welcomed the man with open arms, allowing him to take his pick of who he wanted. The stranger cast his eyes along the crowd, passing by everyone I knew, his gaze nonchalant to all. All that is, except me. He looked straight at me, and stopped. Eyes bloodshot, his mouth lined with dry, cracked vomit, his shirt, blooded, torn and stained. The memory flooded through me. It was then I realized just how blind I had been. The night that changed it all, it had not been emptiness in the boy's eyes. It had been lust! And it was lust again the bore its way inside my soul. The stranger had chosen. I wouldn't be saying goodbye to a friend this time. I would be saying goodbye to everything. The door opened to his building. And a building it would always be, never a home. No place with so much vile hatred seeping out of every crack in the wall, so much murk and gloom pouring out of every shadow, could ever be called a home. He sat in front of me, breathing his horrid breath over my neck, running his fingers down my body, inhaling my breath. I knew straight away, this was never going to end well. He lifted me, I felt weightless and powerless under his grasp. He put his lips around me, and it felt like he drank the very life force from my body. His brown, crusted teeth scraping against me, as his course tongue brought him to the satisfaction he had desired for so long. After what felt like an eternity in the lowest ring of hell, he was done. The door was opened, and for the second time in my life, I was cast out. This time however, there was no remorse. There was nothing. I was resound to my fate.And that's how I arrived here, lying once again in a gutter. This time, I know there will be no retribution. I can see my life flowing from my body. It's coming to an end. If only things had been different. If only we really did have medicinal qualities, then maybe, just maybe, this wouldn't be the end of just one more bottle of Buckfast. Question: Answer the following question based on the above passage:Summarize what, according to the narrator, led to their being in a gutter. Answer: The narrator was deeply connected to the host, a boy who took the narrator in. After one night out, the narrator came back with dried bloody vomit trailing down his shirt, covered in blood, and the next day the host launched into a tirade of words against the narrator and tore the narrator out of its home. Reduced to a vague memory now, the narrator would recall the chilling last moment they had together. It was a night of deep lust that stained their friendship through the lasting and eternal pains of betrayal and lust. Your Feedback:"
Output for error pinpoint model:
This response contains a major coherence error. The sentence is written in a confusing way. The narrator realizes the lust in the boy's eyes, and their last night is horrendous for it. Then, the narrator is cast out again.

Table 11: A topical summarization example prompt and output we used for our error pinpoint trained from from PALM2, where a summarization example is based on a particular question. According to our scoring scheme, one major error corresponds to 2, at 1 to 5 scale. The normalized score is 40 out of 100.

Search Algorithm	Zh-En		En-De		ASQA	Summ
	22	23	22	23	QA	Sum
ZERO-SHOT	67.6	67.3	79.0	77.0	18.3	26.1
ALWAYS ACCEPT (1)	69.3	68.4	79.4	77.5	26.2	27.0
GREEDY UPHILL (1)	69.1	68.8	79.6	77.1	25.5	27.5
SIM. ANNEALING (1)	69.2	68.4	79.7	77.5	25.4	27.5
ALWAYS ACCEPT (5)	69.9	68.3	80.0	78.1	25.4	27.4
GREEDY UPHILL (5)	69.6	68.9	80.1	77.3	25.7	27.4
SIM. ANNEALING (5)	**70.1**	**69.2**	**81.0**	**78.4**	**26.4**	**28.3**

Table 12: We include iterative refinement results from three search algorithms: 1) Always Accept 2) Greedy Uphill 3) Simulated Annealing for 1 iteration and 5 iterations. Different from Table 3, all search algorithms are performed with top-k sampling at each step and we report results on examples that are marked as "errors" by our feedback model.

Metric	en-de			zh-en		
	P	R	F_1	P	R	F_1
AutoMQM (Bison)	0.06	0.64	0.11	0.14	0.21	0.17
Error Pinpoint	0.29	0.20	0.24	0.29	0.36	0.32

Table 13: Character-level precision/recall/F_1 of different reference-free error span tagging models on the full WMT'22 dataset.

LLMMend vs	Win	Neutral	Lose	Win lose ratio
0-SHOT	38%	46%	16%	2.34
IMPROVE	39%	45%	16%	2.44
BLEURT-SCORE-QE	41%	44%	15%	2.79
BLEURT-BINARY-QE	33%	48%	19%	1.76
SCORE-QE	33%	40%	27%	1.23
BINARY-QE	34%	48%	18%	1.84

Table 14: We conduct head-to-head human evaluation on LLMMend against all baselines for single step refinement at WMT22 En-De. We report percentages of win, neutral and lose and win/lose ratio in the table.

Simulated Annealing vs	Win	Neutral	Lose	Win lose ratio
ALWAYS-ACCEPT	38%	38%	24%	1.56
GREEDY UPHILL	31%	47%	22%	1.38

Table 15: We conduct head-to-head human evaluation on our simulated annealing based search algorithm against naive search algorithms at WMT22 En-De. We report percentages of win, neutral and lose and win/lose ratio in the table.