

Wayfinder: An Agentic Ensemble-and-Postcheck System for WMT26 GenMT

Liangying Shao, Xinwei Wu, Yichong Huang, Jifang Wang,
Ruoxi Xu, Ling Shi, Baosong Yang, Linlong Xu

Alibaba Group

Abstract

This paper presents Wayfinder, our submission system for the WMT26 General Machine Translation (GenMT) shared task. GenMT evaluates document-level translation under natural-language instructions, where structural preservation, formatting constraints, and output cleanliness are part of translation quality. Rather than training a new translation model, Wayfinder treats four API-based system outputs as a document-level candidate pool and produces one final hypothesis through a four-stage agentic pipeline. It aggregates source, instruction, metadata, candidates, and a DeepSeek-V4-Pro ranking-then-scoring signal; applies an anonymized pick-or-rewrite agent; and runs a conservative postcheck for severe objective defects. During development, a judge-guided loop compares outputs with a GPT-5.5 baseline, analyzes loss cases, updates prompts and repair constraints, and selectively reprocesses difficult documents. In automatic pairwise evaluation against GPT-5.5, Wayfinder obtains a pooled document-level weighted win rate of 0.6537 and remains above parity in every evaluated direction. The results suggest that candidate-pool translation, constrained agentic selection, conservative repair, and selective judge-guided iteration are practical components for instruction-conditioned shared-task MT.

1 Introduction

Large language models have substantially changed the design space of machine translation (Achiam et al., 2023; Bai et al., 2023; Alves et al., 2024). Beyond improving general translation quality, they make it possible to condition translation on rich, task-specific instructions, moving MT closer to settings where the desired output is defined by both linguistic meaning and downstream use constraints. This shift is reflected in the WMT26 GenMT shared task¹, which evaluates document-

level translation under natural-language instructions and extends prior work on document-level MT toward instruction-conditioned evaluation. In such an instruction-conditioned setting, a system is expected to preserve the meaning of the source document while also satisfying explicit requirements on structure, formatting, register, and output cleanliness, such as retaining markup, placeholders, JSON keys, URLs, and line breaks. These requirements are related to constrained generation, where output validity depends not only on content quality but also on satisfying explicit lexical or structural constraints (Hokamp and Liu, 2017; Rei et al., 2025; Sun et al., 2026). As a result, adequacy and fluency alone are insufficient: a translation that reads naturally may still fail the task if it breaks a required format, drops a protected token, or adds content disallowed by the instruction.

Motivated by this instruction-conditioned setting, our submission, Wayfinder, is built on the observation that current LLM translation systems are strong but complementary. Across language directions, document types, and constraint patterns, the best hypothesis may come from different systems, and even a generally strong model may occasionally violate a required structure or protected token. Rather than training a new translation model, Wayfinder treats several system outputs as a document-level candidate pool and uses an agentic selection-and-rewriting stage to produce one final hypothesis per document. This design echoes earlier MT system-combination, reranking, and minimum Bayes-risk selection work (Kumar and Byrne, 2004; Freitag et al., 2021b), while adapting the idea to LLM-based translation, where performance remains sensitive to language, domain, and prompting setup (Jiao et al., 2023; Cui et al., 2025; Wang et al., 2026).

Wayfinder operationalizes this idea as a four-stage agentic pipeline. First, it constructs a per-document evidence record by aligning the source

¹<https://www2.statmt.org/wmt26/translation-task.html>

document, natural-language instruction, target-language metadata, four candidate translations, and a ranking-then-scoring judge signal. Second, it anonymizes candidate identities and asks a pick-or-rewrite agent to either select the strongest existing hypothesis or synthesize a corrected translation from the candidate pool under an explicit rubric. Third, a conservative postcheck detects severe objective defects, such as broken structure, truncation, repetition, wrong-language residue, or malformed JSON, and applies either deterministic repair or targeted regeneration. Finally, during development, a judge-guided iteration loop treats GPT-5.5 as an anchored baseline and performs multi-round tuning: it compares system outputs against the baseline, identifies judge-underperforming documents, updates prompts and postcheck configurations from the observed failure modes, and reprocesses those documents with the goal of improving beyond the baseline.

As a shared-task system paper, our goal is to make these design choices transparent rather than to claim a new training method. We describe how Wayfinder builds document-level evidence, constrains the agent’s decision space, hides candidate provenance during selection, checks only severe objective failures, and uses judge feedback to tune the system across rounds. We also report internal diagnostics, including a small anonymization ablation, to explain why the submitted configuration was chosen; these analyses are intended to characterize the system before official evaluation rather than to establish general claims about automatic judging or anonymization.

The rest of the paper is organized as follows. Section 2 describes the Wayfinder pipeline. Section 3 presents the data, candidate systems, agent models, and judging setup. Section 4 reports automatic development results and diagnostics. Section 6 summarizes limitations.

2 System Description

2.1 Pipeline Overview

Wayfinder produces exactly one translation for each source document. Let d denote a document, ι_d its instruction, and $\mathcal{C}_d = \{h_1, \dots, h_K\}$ the candidate translations from $K = 4$ base systems. The pipeline has four stages:

1. **Evidence aggregation:** join the source document, instruction, target-language metadata,

four candidate translations, and a unified ranking-then-scoring judge signal by `doc_id`.

2. **Anonymized pick-or-rewrite:** call an LLM agent on anonymized candidates and judge signals; the agent either picks one candidate or rewrites a fused output.
3. **Conservative postcheck:** detect severe objective defects, then either apply a deterministic repair or ask a repair agent to regenerate with the detected problems as constraints.
4. **Judge-guided iteration:** during development, use GPT-5.5 as an anchored baseline selected from the base-system ranking results; compare each round against it, diagnose judge-underperforming documents, update prompts and postcheck configurations, and selectively re-run those documents to improve the system beyond the baseline.

Figure 1 summarizes the four stages and the development-time feedback loop.

2.2 Evidence Aggregation

The first stage converts heterogeneous inputs into a single per-document evidence record. For each `doc_id`, this record contains the source document, natural-language instruction, target-language metadata, four candidate translations, and a unified preference signal from a ranking-then-scoring judge. The goal is to provide the downstream agent with a document-local view of the candidate pool rather than a set of disconnected model outputs.

In our experiments, this judge is instantiated with DeepSeek-V4-Pro (DeepSeek-AI et al., 2026). It evaluates all four candidates in one call, conditioned on the same source document and instruction. It first produces a global ranking, allowing ties when candidates are indistinguishable under the task requirements, and then assigns scalar scores consistent with that ranking. Compared with independent pairwise judgments, this formulation keeps the ranking and score evidence calibrated within the same four-way comparison. The resulting signal summarizes both the judge’s preferred hypothesis and the confidence structure of the candidate pool, including cases where candidates are close or complementary.

All inputs are aligned by `doc_id`. Judge outputs that are missing or malformed are preserved as missing fields, not converted into negative evidence. In such cases, the pick-or-rewrite agent

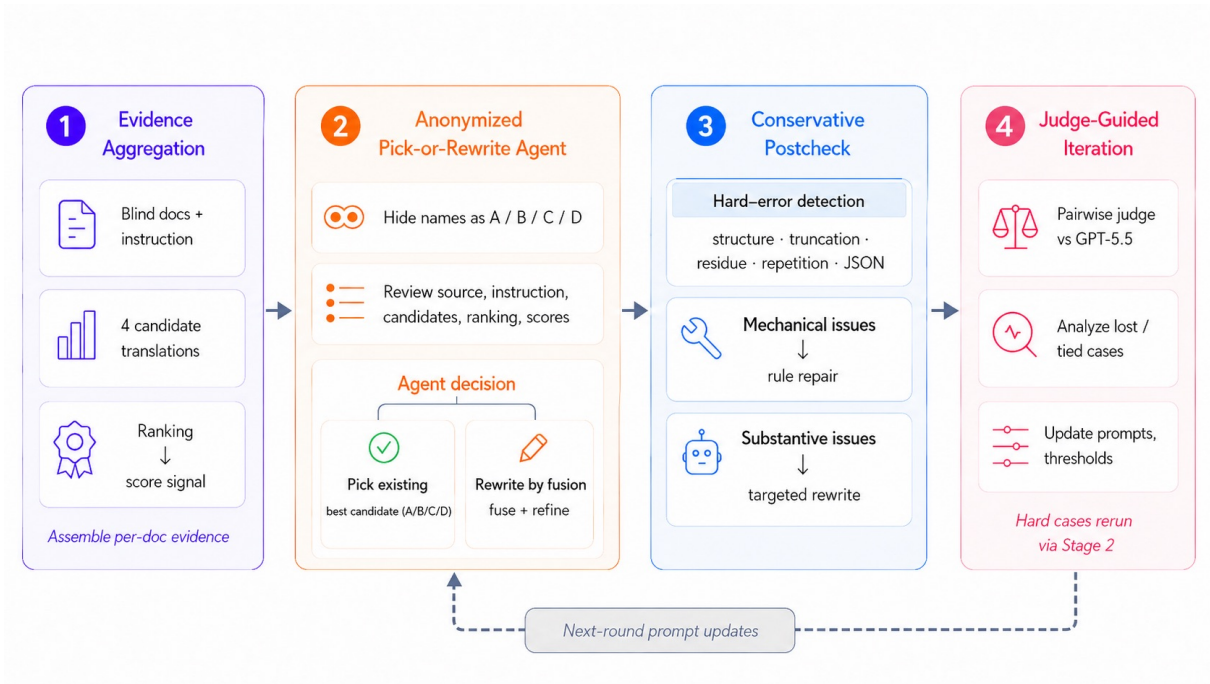


Figure 1: Overview of Wayfinder, our WMT26 GenMT submission system. The pipeline aggregates per-document evidence from four candidate translations, uses an anonymized pick-or-rewrite agent and a conservative postcheck, and applies judge-guided multi-round tuning during development.

still receives the source, instruction, metadata, and candidate translations; the absent judge signal simply removes one source of preference evidence. We also preserve row-level target-language metadata instead of deriving it only from file names, since some WMT26 directions contain fine-grained target-language codes or share target scripts across different source languages.

2.3 Anonymized Pick-or-Rewrite Agent

The second stage converts the aggregated evidence record into a single translation hypothesis. For each document, the agent receives the source, instruction, four anonymized candidate translations, and the ranking-then-scoring signal described above. Its output is constrained to a small decision schema:

```

{
  "chosen_base": "A|B|C|D",
  "action": "pick|rewrite",
  "final_translation": "...",
  "issues": ["..."],
  "reason": "..."
}

```

The action field separates two kinds of decisions. When the agent selects pick, the final translation is required to be identical to the chosen candidate, preserving a high-quality hypothesis without unnecessary modification. When it selects rewrite,

the chosen candidate is treated as the primary draft, but the agent may incorporate corrections from the other candidates. Thus, rewriting is used only when fusion or targeted correction is expected to improve the final output, for example by combining the structural fidelity of one candidate with the terminology, completeness, or fluency of another.

Candidate anonymization. Before the agent call, Wayfinder masks the provenance of the four candidate systems. For each document, the real system names are mapped to neutral labels A/B/C/D by a stable document-specific permutation. The same anonymized labels are used consistently in the candidate block, ranking, score fields, and textual references to candidate identity. The real mapping is retained outside the prompt for analysis, but it is not exposed to the decision agent.

This design is intended to make the agent compare candidates using document-local evidence rather than model identity. Because the label assignment changes across documents while remaining reproducible for a given document, any residual preference for a particular letter is not consistently tied to a particular base system.

Evaluation rubric. The agent prompt uses an ordered rubric to make the decision criteria explicit. The criteria are loosely aligned with error-based

MT evaluation, Direct Assessment, and MQM-style analysis, while being simplified for a system prompt rather than a human annotation protocol (Graham et al., 2013; Lommel et al., 2014; Freitag et al., 2021a). The first two are hard constraints:

1. Instruction adherence: satisfy register, formatting, placeholder, URL, hashtag, markup, JSON-key, and output-only requirements in the task instruction.
2. Structural full match: preserve JSON shape, HTML tag tree, line count, placeholders, punctuation, whitespace, and newlines.
3. Fidelity: avoid omissions, additions, and hallucinations.
4. Terminology, named entities, numbers, dates, and units.
5. Grammar: pronouns, gender, number, tense, and agreement.
6. Register and formality.
7. Target-language fluency.
8. Output cleanliness: no source-language residue, meta-commentary, or stray code fences.

The ranking and score signals are presented as advisory evidence, not as binding decisions. The agent is instructed to override them when direct inspection of the source, instruction, and candidate translations reveals a structural or adequacy problem. The prompt also includes a pass-through exception for garbled or non-linguistic inputs: a candidate identical to the source is not treated as erroneous by itself when translation is not well-defined.

Robustness to malformed decisions. Although the agent is prompted to return a schema-constrained decision, malformed outputs can occur. If the chosen candidate label is invalid, the controller attempts to recover it by matching the final translation against the candidate strings. If the decision cannot be parsed or recovered, the system falls back to the top candidate from the ranking-then-scoring judge when available, and otherwise to GPT-5.5. This ensures that every document receives a final hypothesis.

Postcheck signal	Repair policy
Length ratio	Flag likely truncation or over-generation; regenerate when the mismatch reflects content loss or expansion.
Repeated chunk	Flag repeated spans not licensed by the source; regenerate when the pattern indicates looping.
Line-count mismatch	Flag broken line structure; regenerate unless the difference is limited to blank lines.
Latin residue	Flag unexpected Latin text in non-Latin targets; regenerate when it reflects wrong-language residue.
JSON validity	Flag malformed structured output; repair mechanically when safe, otherwise regenerate.
Empty output	Flag empty or extremely short hypotheses; regenerate.

Table 1: Postcheck signals and repair policies.

2.4 Conservative Postcheck

The third stage is a conservative safeguard for severe objective failures in the current hypothesis. It is not intended to be a general post-editor: open-ended stylistic improvement is avoided because unnecessary rewriting can introduce regressions. Instead, the postcheck asks two narrow questions. First, does the hypothesis exhibit a hard failure that would make the output unusable under the instruction? Second, if such a failure is present, what is the least invasive repair that can address it? This focus on severe failures is motivated by known hallucination, omission, and instability phenomena in neural and LLM-based MT systems (Raunak et al., 2021; Guerreiro et al., 2023).

The detector combines lightweight rules with a repair controller. The rules flag hypotheses that are likely to contain structural or content-level defects, such as truncation, repetition, wrong-language residue, broken JSON, or line-count mismatch. Table 1 summarizes the rule triggers and their typical repair actions.

For each flagged hypothesis, the repair controller chooses among three outcomes. When the issue is purely mechanical and the correction is unambiguous, the system applies a deterministic rule-based repair; for example, it can restore missing code-fence boundaries or repair a recoverable delimiter mismatch in an otherwise intact structured output. When the issue reflects a substantive translation failure, such as hallucination, large omission, wrong-language output, repeated generation,

or a structure mismatch that loses content, the system calls a repair agent. The repair agent receives the source, instruction, current hypothesis, original candidates, and detected issue labels, and regenerates a corrected translation under those constraints. When the flag appears to be a false positive, the hypothesis is kept unchanged.

The repair prompt is deliberately restrictive. It permits edits only for severe failures, including large truncation or omission, hallucination, broken JSON/HTML/line structure, repeated generation, wrong-language output, empty or extremely short output, or incorrect key entities, numbers, and dates. It explicitly forbids tone polishing, synonym substitution, and minor register adjustments. If a repair decision cannot be parsed, the system keeps the original hypothesis rather than risking an uncontrolled change.

Postcheck is applied iteratively by re-scanning the current output after each repair round. In development, this iterative rescanning converged within a small number of repair rounds.

2.5 Judge-Guided Iteration

The final stage uses automatic judgments as a development signal for multi-round system tuning. Section 4.1 shows that Qwen3.7-Max and GPT-5.5 form a near-tied top group among the four base systems. We use GPT-5.5 as the anchored baseline because it provides a stable strong single-system reference point: Qwen3.7-Max has a slightly higher Top-1 share, while GPT-5.5 has the higher strict-first rate. Since Qwen3.7-Max also serves as the inference model for the agentic decision and repair components, using GPT-5.5 as the baseline keeps the evaluation anchor distinct from the model that constructs Wayfinder outputs. The baseline is therefore not treated as a gold reference or as the uniquely best base system; it is a fixed anchor for measuring whether prompt and postcheck changes improve the pipeline.

At the end of each development round, the current Wayfinder output is compared against the GPT-5.5 baseline at the document level with randomized A/B presentation. Documents with poor judge outcomes, including losses and selected low-confidence ties, are selected for error analysis. For these documents, the analysis agent examines the source, instruction, system output, baseline output, agent issue codes, postcheck flags, and ranking-then-scoring evidence. The analysis is used to identify recurring failure modes, such as

over-aggressive rewriting, failure to preserve structure, weak handling of a particular genre, or over-reliance on a high-scoring candidate.

The next round then updates the system configuration rather than the underlying translation models. Possible updates include revising the pick-or-rewrite prompt, sharpening decision instructions, adding or tightening postcheck rules, or changing repair-agent constraints. Only the selected underperforming documents are reprocessed with the updated configuration, while documents with satisfactory judge outcomes are carried forward. This selective procedure keeps the loop practical while still using judge feedback to push the assembled system beyond the anchored baseline.

During development, we keep the output of each round. For each `doc_id`, the final development set can then use the strongest hypothesis observed so far according to the development judge. We report gains from this procedure only as development diagnostics, since this selection is part of system construction rather than a separately controlled experiment.

3 Experimental Setup

3.1 Task and Data

We participate in 28 language directions selected from the WMT26 GenMT blind test set. For the selected directions, we parse the raw JSONL blind-test records and group them by source and target language. The resulting submission and automatic evaluation cover 19,880 documents.

3.2 Base Candidate Systems

The candidate pool contains four API-based translation systems: Gemini 3.1 Pro,² Gemini 3.5 Flash,³ GPT-5.5,⁴ and Qwen3.7-Max.⁵ GPT-5.5 is also used as the anchored baseline in our automatic pairwise evaluation, following the ranking-then-scoring results reported in Section 4.1. This gives a fixed near-best comparison point across language directions while still allowing Wayfinder to select or rewrite from any of the four candidates.

²<https://deepmind.google/models/gemini/pro/>

³<https://deepmind.google/models/model-cards/gemini-3-5-flash/>

⁴<https://developers.openai.com/api/docs/models/gpt-5.5>

⁵<https://qwen.ai/blog?id=qwen3.7>

3.3 Agent and Judge Models

Qwen3.7-Max serves as the inference model for the agentic components, namely the pick-or-rewrite agent, the postcheck repair agent, and the round-analysis agent; it is also included in the base candidate pool. To keep automatic evaluation independent of this agentic decision model, the ranking-then-scoring judge and the automatic pairwise evaluation judge are instantiated with DeepSeek-V4-Pro (DeepSeek-AI et al., 2026). Structured stages are constrained to produce JSON outputs matching the schemas described in Section 2.

3.4 Automatic Judging Protocol

Automatic development evaluation uses DeepSeek-V4-Pro to compare the system output (SYS) against the GPT-5.5 baseline (REF), aligned by doc_id. The protocol is intended as an internal diagnostic rather than a replacement for human evaluation, echoing prior concerns about MT evaluation reliability and judge bias (Zheng et al., 2023; Kim, 2025). The judge sees randomized A/B presentation order and returns one of

SYS $\gg$ REF, SYS > REF, TIE,
REF > SYS, REF $\gg$ SYS.

We report a weighted win rate. The metric gives 3 points to a strong SYS win, 1 point to a slight SYS win, 0.5 points to a tie, and 0 points to losses, normalized by the maximum possible score. This metric gives more weight to decisive improvements and strong failures.

These automatic scores were used for development and analysis as internal measurements of the system configuration. We use pairwise LLM judging as a development diagnostic rather than standard reference-based metrics such as BLEU, METEOR, TER, chrF, BERTScore, BLEURT, or COMET, since the current task also stresses instruction adherence and hard format constraints (Papineni et al., 2002; Banerjee and Lavie, 2005; Snover et al., 2006; Popovic, 2015; Zhang et al., 2020; Sellam et al., 2020; Rei et al., 2020).

4 Results

4.1 Base-System Ranking Results

We first summarize the ranking-then-scoring results for the four base systems, using DeepSeek-V4-Pro as the automatic judge. This analysis motivates the document-level selection design and determines the anchored baseline used in the final pairwise

System	Top-1	Strict first
Qwen3.7-Max	27.3	17.7
GPT-5.5	27.1	19.6
Gemini 3.5 Flash	26.6	16.6
Gemini 3.1 Pro	19.0	10.7

Table 2: Aggregate ranking-then-scoring results for the four base systems over the submitted directions. Values are percentages; tied Top-1 positions are fractionally allocated.

evaluation. Table 2 reports aggregate rank-based metrics over the 28 submitted directions. Because raw score values are not directly comparable across all scoring procedures, we aggregate Top-1 share and strict-first rate rather than average scores.

The results show that no single base system dominates the candidate pool. Qwen3.7-Max and GPT-5.5 form a near-tied top group: Qwen3.7-Max has the highest Top-1 share by a very small margin, while GPT-5.5 has the highest strict-first rate. Given this mixed evidence, we do not treat the ranking table as identifying a unique best base system. For the pairwise evaluation below, we use GPT-5.5 as REF as a fixed, strong single-system anchor for measuring whether Wayfinder improves over a near-best candidate. At the same time, it keeps the reference system distinct from Qwen3.7-Max, which serves as the inference model for Wayfinder’s agentic decision and repair stages. The baseline selection does not constrain the pipeline itself, which may still select or rewrite from any of the four candidates on a document-by-document basis.

4.2 Final Pairwise Results against GPT-5.5

Table 3 reports automatic pairwise judging of Wayfinder against GPT-5.5 over the 28 submitted directions. Under this protocol, every judged direction remains above parity under weighted win rate. The per-direction scores range from 0.5283 to 0.8668, and pooling all document-level judgments gives an overall weighted win rate of 0.6537.

The highest scores occur in directions with many decisive SYS wins, such as en-ind_Latn and cs-vie_Latn. Directions with high tie rates, such as tr-en and en-es_ES, remain close to the GPT-5.5 baseline. Strong-loss rates are higher in a small number of directions, especially en-lij_Latn, zh-jpn_Jpan, and en-sme_Latn, indicating that the aggregate gains are not uniform across directions.

Direction	<i>n</i>	SYS>>REF	SYS>REF	TIE	REF>SYS	REF>>SYS	Wtd.
en-ind_Latn	198	53.0	26.8	3.0	12.1	5.1	0.8668
cs-vie_Latn	315	39.4	34.0	3.5	16.5	6.7	0.8008
cs-deu_Latn	315	30.5	42.2	6.3	14.9	6.0	0.7908
en-tha_Thai	198	42.4	31.3	4.5	12.1	9.6	0.7884
en-jpn_Jpan	198	43.9	26.3	3.0	19.2	7.6	0.7861
en-zho_Hant_TW	198	38.9	34.3	3.0	15.2	8.6	0.7824
en-lld_Latn	198	50.5	11.6	18.2	7.6	12.1	0.7646
cs-ukr_Cyrl	315	29.5	38.1	6.0	18.4	7.9	0.7414
en-zh_CN	945	15.2	44.0	20.6	17.1	3.0	0.7335
en-sme_Latn	198	47.5	15.7	10.1	9.6	17.2	0.7115
en-hye_Armn	198	32.3	28.3	8.6	20.2	10.6	0.6970
en-bel_Cyrl	198	28.8	31.3	10.6	18.2	11.1	0.6840
en-kaz_Cyrl	198	27.3	27.8	16.7	18.7	9.6	0.6788
en-arz	348	20.7	33.3	22.1	14.7	9.2	0.6664
en-ru	2115	13.4	34.1	32.1	15.4	4.9	0.6619
en-pl_PL	596	8.1	36.6	36.1	17.4	1.8	0.6576
en-ukr_Cyrl	198	25.3	37.9	3.5	20.7	12.6	0.6566
en-ko_KR	2078	13.4	25.6	41.8	15.8	3.5	0.6471
en-de_DE	1861	7.7	30.3	48.3	11.3	2.4	0.6452
en-cs	2165	11.4	32.7	34.4	16.9	4.7	0.6357
en-ar_AR	150	2.0	42.7	39.3	14.0	2.0	0.6327
zh-jpn_Jpan	188	36.2	21.3	5.3	17.6	19.7	0.6256
ko-en_US	3536	6.7	31.7	42.7	16.1	2.8	0.6142
en-es_ES	917	2.8	21.0	70.8	4.6	0.8	0.6058
en-is	543	14.7	20.8	44.4	12.5	7.6	0.6032
en-et_EE	1115	8.3	23.9	51.0	13.2	3.6	0.6007
en-lij_Latn	198	31.3	19.2	12.6	11.1	25.8	0.5578
tr-en	200	2.0	10.5	79.0	7.5	1.0	0.5283
All documents	19880	14.5	30.3	35.8	14.7	4.8	0.6537

Table 3: Automatic pairwise results against GPT-5.5. Verdict columns are percentages of document-level judgments, and Wtd. denotes the weighted win rate. The final row pools judgments across all submitted directions.

Direction	Name exposed	Anonymized	Δ
en-kaz_Cyrl	0.6096	0.6788	+6.9 pts
en-zh_CN	0.7241	0.7335	+0.9 pts
en-ukr_Cyrl	0.6680	0.6566	-1.1 pts

Table 4: Effect of candidate-identity anonymization on weighted win rate in three development directions.

4.3 Development Diagnostics

We use the following diagnostics to explain several design choices in the submitted system. They are not intended as standalone controlled experiments; rather, they summarize observations that guided the final configuration of Wayfinder. Table 4 compares two candidate-identity settings on three development directions: a name-exposed configuration, where the agent saw the real system names, and the submitted anonymized configuration, where candidates were presented only as A/B/C/D.

The anonymized setting is not uniformly better in aggregate score, but it reduces the chance that the decision agent relies on model provenance rather than document-local evidence. We therefore treat anonymization as a conservative design

choice, especially for preventing severe selection errors, rather than as an independent source of consistent score gains.

The postcheck diagnostics support a similarly conservative interpretation. For example, on en-zh_CN, it changed 31 of 945 outputs (3.3%), including deterministic restoration of dropped code fences and targeted repairs for hard structural defects. Across development runs, the postcheck was mainly activated by mechanically verifiable defects, such as missing code fences, broken structured output, line-count mismatch, repetition, or empty hypotheses. This behavior is consistent with its intended role: it is a safeguard against invalid or clearly defective translations, not a general rewriting stage.

Finally, judge-guided iteration was used to tune the system configuration rather than the underlying translation models. Documents with satisfactory pairwise outcomes were carried forward, while documents with losses or low-confidence ties were reprocessed after prompt or postcheck updates. In early development runs, this selective procedure reduced the number of documents sent to later

rounds by roughly 60%, and cross-round selection improved the weighted score by about 2–5 points over the best individual round. We report these numbers as development diagnostics because the iteration loop is part of the system-building process rather than an isolated algorithmic result.

5 Conclusion

We presented Wayfinder, our submission to the WMT26 GenMT shared task. Wayfinder is organized as a four-stage pipeline: it aggregates document-level evidence from four candidate translation systems and a ranking-then-scoring judge, applies an anonymized pick-or-rewrite agent, performs conservative postcheck for severe objective errors, and uses judge-guided iteration during development to refine prompts and repair behavior. This design targets instruction-conditioned document translation, where preserving structure, formatting, and task-specific constraints is part of translation quality rather than a separate post-processing concern.

In internal automatic pairwise evaluation against GPT-5.5, Wayfinder remains above parity in all 28 submitted directions, with an overall pooled weighted win rate of 0.6537. These results support the utility of document-level candidate selection, constrained rewriting, severe-error postchecking, and selective judge-guided iteration for instruction-conditioned shared-task MT when strong base systems are complementary rather than uniformly ordered.

6 Limitations

Wayfinder has three main limitations. First, the pipeline is relatively expensive: each document starts from four API-based candidate translations and may additionally require ranking, agentic selection or rewriting, postcheck, and selective reprocessing. This design is suitable for an offline shared-task submission, but it is not optimized for low-latency or cost-constrained deployment. Second, Qwen3.7-Max is used both as one candidate translation system and as the inference model for the pick-or-rewrite, repair, and round-analysis agents. Although candidate anonymization and the use of a separate DeepSeek-V4-Pro judge reduce direct provenance effects, this concentration may still introduce model-family-specific preferences into the final system behavior. Third, Wayfinder depends on the diversity and quality of the four-

system candidate pool; errors shared by all candidates are difficult for a selection-and-rewriting pipeline to correct reliably. Future work should explore more efficient candidate filtering and judging, broader candidate pools, and stronger safeguards against model-specific decision bias.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Duarte M. Alves, José Pombal, Nuno Miguel Guerreiro, Pedro Henrique Martins, João Alves, M. Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *CoRR*, abs/2402.17733.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *CoRR*, abs/2309.16609.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.
- Menglong Cui, Pengzhi Gao, Wei Liu, Jian Luan, and Bin Wang. 2025. [Multilingual machine translation with open large language models at practical scale: An empirical study](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, New Mexico. Association for Computational Linguistics.
- DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, and 1 others. 2026. [DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence](#). *Preprint*, arXiv:2606.19348.
- 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, David Grangier, Qijun Tan, and Bowen Liang. 2021b. Minimum bayes risk decoding with neural metrics of translation quality. *arXiv preprint arXiv:2111.09388*.

- Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41.
- Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and Andre F. T. Martins. 2023. Hallucinations in Large Multilingual Translation Models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.
- 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*, pages 1535–1546.
- Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? A Preliminary Study. *arXiv preprint arXiv:2301.08745*.
- Ahrii Kim. 2025. [RUBRIC-MQM : Span-level LLM-as-judge in machine translation for high-end models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 147–165, Vienna, Austria. Association for Computational Linguistics.
- Shankar Kumar and William Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 169–176.
- Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics. *Tradumatica*, 12:455–463.
- 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.
- Maja Popovic. 2015. chrF: Character n-gram F-score for Automatic MT Evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.
- Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The Curious Case of Hallucinations in Neural Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183.
- Ricardo Rei, Nuno Miguel Guerreiro, José Pombal, João Alves, Pedro Teixeira, M. Amin Farajian, and André F. T. Martins. 2025. Tower+: Bridging generality and translation specialization in multilingual llms. *CoRR*, abs/2506.17080.
- 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*, pages 2685–2702.
- Thibault Sellam, Dipanjan Das, and Ankur P. 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.
- Matthew Snover, Bonnie Dorr, Richard 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*, pages 223–231.
- Mingrui Sun, Mao Zheng, Zheng Li, and Mingyang Song. 2026. Ifmtbench: A comprehensive benchmark for multilingual translation instruction following. *arXiv preprint arXiv:2605.28218*.
- Hao Wang, Linlong Xu, Heng Liu, Yangyang Liu, Xiaohu Zhao, Bo Zeng, Liangying Shao, Yichen Dong, Xinwei Wu, Jiang Zhou, Tianyu Dong, Xiangxiang Zeng, Longyue Wang, and Weihua Luo. 2026. [M²PO: Multi-perspective multi-pair preference optimization for machine translation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10315–10336, San Diego, California, United States. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.