Ehime-U System for the WAT 2026 Patent Claims Evaluation Task: LLM-Based Translation Error Prediction Using Word Alignments

Wang Xu Yin **Isao Goto** **Takashi Ninomiya**
Ehime University
wang@ai.cs.ehime-u.ac.jp {goto.isao.fn,ninomiya.takashi.mk}@ehime-u.ac.jp

Abstract

This paper describes the Ehime-U system submitted to the Japanese–English Patent Claims Translation/Evaluation Task at WAT 2026. Our system predicts translation errors by providing an LLM with word-level correspondences between source and target claims. We compare six settings that differ in how these correspondences are obtained and whether they are used: four Unbalanced Optimal Transport (UOT)-based alignment methods (Forward, Reverse, AND, and OR), one alignment generated by LLM, and a baseline without alignment. We discuss how the choice of word correspondence affects error prediction. Against the gold reference annotations, under the hyperparameter settings used in this study, providing word correspondences improves error-detection F1 by up to 3.7 points over the no-alignment baseline (0.512 vs. 0.475). The improvement is category-specific: recall increases for alignment-dependent error categories but not for fluency errors. Comparing the two approaches to obtaining correspondences, we find that LLM-based alignment covers substantially more source tokens than the UOT alignments and achieves a higher F1 score than the best UOT setting under the hyperparameter settings used in this study (0.512 vs. 0.486).

1 Introduction

The Ehime University team (Ehime-U) participated in the Japanese–English Patent Claims Translation/Evaluation Task at the 13th Workshop on Asian Translation (WAT 2026) (Nakazawa et al., 2026). This paper describes the system that produced our submitted results. Patent claim translation requires both terminological precision and structural fidelity (Buma et al., 2025; Nakazawa et al., 2025). Errors such as omissions, additions, and mistranslations can change the legal scope of a claim (Li et al., 2019), making accurate error annotation critical (Lommel et al., 2013). Most existing systems prompt an LLM to compare the source and target directly (Kocmi and Federmann, 2023; Lu et al., 2024). In contrast, our system provides the LLM with explicit word-level correspondences as additional input when predicting translation errors.

Our motivation is that many patent-claim errors are alignment-level phenomena (Li et al., 2019; Arase et al., 2023). A source content word with no target counterpart is likely an omission candidate; a target word with no source counterpart is likely an addition candidate; a source word aligned to the wrong target word is likely a mistranslation candidate. Explicit correspondences make these patterns visible to the model.

Our contributions are as follows:

- We empirically evaluate the effect of using estimated word correspondences in LLM-based translation error prediction. Against the gold reference annotations, under the hyperparameter settings used in this study, providing explicit word correspondences improves error-detection F1 over the no-alignment baseline, and the LLM-based alignment, which covers substantially more source tokens than the UOT alignments, achieves the best overall F1 (0.512 vs 0.475 for the baseline). Moreover, the improvement is selective: recall increases for alignment-dependent error categories (accuracy and terminology consistency), while recall for fluency errors remains unchanged. These results show that, in Japanese–English patent claim error prediction, the proposed method using word correspondences is effective for some error categories such as omissions and mistranslations.
- We compare two ways of estimating word correspondences: unbalanced optimal transport (Cuturi, 2013; Arase et al., 2023) using scores produced by an LLM and LLM-based alignment estimation. We find that

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 134–144
November 9, 2026 ©2026 Association for Computational Linguistics

LLM-based alignment achieves the highest F1 (0.512), followed by the best UOT setting, UOT-AND (0.486). These results indicate that, under the hyperparameter settings used in this study, the LLM we use estimates word correspondences more appropriately than UOT.

2 System Description

Our system consists of three stages: tokenization, word alignment, and error prediction.

2.1 Tokenization

- Japanese: MeCab in `-Owakati` mode (Kudo et al., 2004).
- English: Moses tokenizer (Koehn et al., 2007).

These token IDs are the atomic units for downstream alignment and error annotation.

2.2 Word Correspondence

For each source–target sentence pair, we compute a set of word links. A word link is a correspondence from a source token to one or more target tokens. We use word links as additional input to the error-prediction stage: they indicate which source and target tokens correspond to each other.

We obtain word links with two different methods. The first estimates a score for each token pair and solves an unbalanced optimal transport (UOT) problem on that score. The second asks an LLM to predicted the alignment directly. We describe the two methods below.

Unbalanced Optimal Transport-based Alignment

We estimate word correspondence by first computing a score for each source–target token pair, and then solving a UOT problem on that score. We use two alternative scoring methods: an LLM-based semantic score (the main setting) and a character-based lexical score (a lightweight fallback).

The semantic score is assigned by an LLM. We present the tokenized source and target as local token IDs and ask the LLM to return a sparse list of plausible token pairs with a score $s \in [0, 1]$ (1.0 = exact translation, 0.7 = strong candidate, 0.4 = weak candidate; unlisted pairs are treated as no correspondence). The full prompt is listed in Appendix A.1. We convert the scores into a dense ground cost with $C[i][j] = 1 - s$ for listed pairs and $C[i][j] = 1$ for unlisted pairs. Because the score is assigned by the LLM based on meaning rather than character overlap, it provides a meaningful signal for Japanese–English alignment.

As a lightweight fallback, we also compute a character-based lexical score

$$\begin{aligned}\mathrm{sim}_{\mathrm{lex}}(s_i, t_j) = \max(&0.5 \cdot \mathrm{Jacc}(s_i, t_j) \\ &+0.5 \cdot \mathrm{Lev}(s_i, t_j),\ \mathrm{Syn}(s_i, t_j))\,,\end{aligned} \tag{1}$$

where $\mathrm{Jacc}(s_i, t_j)$ is the Jaccard similarity of the character n-gram sets of the two tokens ($n = 3$),

$$\mathrm{Jacc}(s_i, t_j) = \frac{|G(s_i) \cap G(t_j)|}{|G(s_i) \cup G(t_j)|}$$

with $G(x)$ the set of character n-grams of token x; $\mathrm{Lev}(s_i, t_j)$ is the normalized Levenshtein similarity

$$\mathrm{Lev}(s_i, t_j) = 1 - \frac{\mathrm{dist}(s_i, t_j)}{\max(|s_i|, |t_j|)}$$

with $\mathrm{dist}(\cdot, \cdot)$ the Levenshtein edit distance (Levenshtein, 1966); and $\mathrm{Syn}(s_i, t_j) = 0.9$ if the pair appears in a small Japanese–English function-word table and 0 otherwise. Because Japanese and English use almost disjoint character sets, this score is only meaningful for numerals, Latin loanwords, and function words.

The semantic score is the main setting. When the LLM is unavailable or returns an empty list of token pairs, we fall back to the character-based lexical score, which is computed entirely offline. Unless otherwise noted, all experiments in this paper use the semantic score.

Whichever score is used, we convert it into a non-negative ground cost

$$C[i][j] = 1 - \mathrm{score}(s_i, t_j) \in [0, 1], \tag{2}$$

where $\mathrm{score}(s_i, t_j)$ denotes either the semantic score s or the lexical score $\mathrm{sim}_{\mathrm{lex}}(s_i, t_j)$. Both scores are in $[0, 1]$, so the resulting cost is also in $[0, 1]$. A smaller cost means a closer correspondence.

Following Cuturi (2013) for the entropy-regularized formulation, Liero et al. (2018) and Chizat et al. (2018) for the marginal relaxation with KL penalties, and Arase et al. (2023) for the unbalanced transport formulation for word alignment,

we then solve the entropy-regularized unbalanced transport problem

$$\min_{P}\Big\{\langle P, C\rangle - \varepsilon H(P) + \tau\big(\mathrm{KL}(P\mathbf{1} \,\|\, \mathbf{a}) + \mathrm{KL}(P^{\top}\mathbf{1} \,\|\, \mathbf{b})\big)\Big\}, \quad (3)$$

where $P \in \mathbb{R}^{n\times m}_{\geq 0}$ is the transport plan (n and m are the numbers of source and target tokens, respectively), C is the cost matrix defined in Equation 2, $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ are the uniform marginal distribution vectors for the source and target tokens, respectively. Specifically, their components are defined as $a_i = 1/n$ for $i = 1, \ldots, n$ and $b_j = 1/m$ for $j = 1, \ldots, m$, ε is the entropy regularization strength, τ is the marginal relaxation strength, $H(P)$ is the entropy of the plan, and $\mathrm{KL}(x \,\|\, y)$ is the Kullback–Leibler divergence. $P\mathbf{1} \in \mathbb{R}^n$ and $P^{\top}\mathbf{1} \in \mathbb{R}^m$ are the row and column sums of P. Their values are given in Section 3.

Each source row $P[i, :]$ is row-normalized into a distribution over target tokens. A target token j is linked to source token i when the normalized mass is at least `link_threshold`; at most `max_links_per_src` target tokens are kept per source token. A source row whose total mass is below `row_mass_floor` is treated as having no correspondence. The values of these three parameters are given in Section 3.

From the same cost matrix we derive four alignments. Forward (source→target) and Reverse (target→source) are obtained by solving UOT on C and on $C^{\top}$, respectively. AND is the strict intersection of the Forward and Reverse link sets, and OR is their union.

LLM-based Alignment

We use an LLM to estimate word correspondence. In this setting, the LLM is given the tokenized source and target claims and is asked to produce span boundaries and a token-level alignment in a single request. The prompt used for this alignment is listed in Appendix A.2. The specific model used is described in Section 3.2.

2.3 Translation Error Prediction Using Word Alignment

In the error prediction stage, we use an LLM to predict translation errors. Each input to the LLM consists of the tokenized source claim, the tokenized target claim, the word correspondence list, and the token IDs of both sides. Each claim is processed independently: the model receives only the current claim, and no information from other claims of the same patent is provided. The LLM then predicts translation errors, each containing five parts: the side (`input` or `output`), the span (source side uses `source_token_ids`, target side uses `target_token_ids`), the MQM type (e.g., 誤訳, 訳抜け, 文法) (Lommel et al., 2013), and the severity (小 or 大).

The full prompt is listed in Appendix A.3. It contains the word correspondence list and asks the model to examine the source–target correspondence item by item, using each aligned pair as reference and referring to the definitions of the MQM error types(e.g., 誤訳, 訳抜け, 文法), following the definitions provided in the task Web page[1], and finally to predict the errors present in the source and target. The prompt also contains the complete token ID lists for both sides, allowing the model to indicate the span of each error by token ID. We then map these IDs back to character offsets: that is, we convert the token IDs produced by the model into a character range in the corresponding text, taking the minimum start and the maximum end across the tokens in the group as the character span of the error.

The prompt additionally includes one few-shot example drawn from the training data, to convey the expected output format. The example is a gold-annotated block containing a source claim, a target claim, and a list of MQM errors, rendered in the same format as the expected output:

```
SOURCE: <source claim>
TARGET: <target claim>
ERRORS: [{"side": ..., "text": ...,
"error": ..., "severity": ...}, ...]
```

The LLM returns a single JSON object with a quality score and a list of errors. Each error is specified by side (`input` or `output`), token IDs, verbatim span text, MQM error type, and severity (小 or 大). We parse and normalize the JSON, and map the token IDs back to character offsets. The resulting annotations are written in the same format as the training data.

2.4 Baseline Translation Error Prediction without Using Word Alignment

In the no-alignment baseline, no word correspondence is provided. An LLM receives only the

[1] https://sites.google.com/view/pat-claims-trans-2026/

source claim, the target claim, and the token ID lists, and must infer errors from the raw source–target comparison. Apart from the instructions regarding the use of word correspondence, the baseline follows the same procedure as described in Section 2.3, using the same few-shot example and output format. The complete prompt used for the no-alignment baseline is provided in Appendix A.3.1.

3 Experimental Setup

3.1 Data

We use test-set files containing Japanese patent claims and their English translations, provided by the Japanese–English Patent Claims Translation Task at WAT 2026. The test set contains 10 claims for JA–EN and 12 claims for EN–JA.

In the LLM-based error prediction stage, the few-shot example is drawn from a single gold-annotated file in the training data:

```
ja-en_sys1_02_WO2024241416.json
```

This file contains gold MQM annotations for one Japanese – English patent claim. We use the first annotated block from this file as the few-shot example.

3.2 Models and Parameters

This section summarizes which model is used for which purpose and lists the parameters of each model.

For word alignment, the semantic score between source and target tokens is computed by an LLM. We use use OpenAI GPT-6 Astra via its API for this scoring step, with reasoning effort `low` and temperature 0; the full prompt is listed in Appendix A.1. Alternatively, the LLM can produce span boundaries and a token-level alignment directly in a single request, as described in Appendix A.2.

For error prediction, we use OpenAI GPT-6 Astra via its API, with reasoning effort `low` to reduce latency, and temperature 0 for deterministic output.

For UOT, we set the entropy regularization strength to $\varepsilon = 0.05$, which controls how smooth the transport plan is.[2] We set the marginal relaxation strength to $\tau = 0.05$, which controls how strongly the source and target marginals are enforced.[3] The link threshold is set to 0.3, the minimum normalized transport mass required to emit a word link. The row mass floor is set to 0.0, effectively disabling this filter. The maximum number of links per source token is set to 3.

[2] Smaller values make the plan closer to a hard assignment; larger values make it smoother and easier to solve.

[3] Smaller values relax the marginals more, so a source token with no counterpart can drop its mass; larger values approach balanced optimal transport.

System Name	Submitted Result Folder
LLM	`GPT6`
BASELINE	`GPT6-noAlignment`
UOT-AND	`AND`
UOT-OR	`OR`
UOT-FORWARD	`Forward`
UOT-REVERSE	`Reverse`

Table 1: Mapping between submitted result folder names and system names used in this paper.

3.3 System Names

We compare six settings.

LLM Uses word correspondences estimated by an LLM and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-AND Estimates word correspondences using the strict intersection (AND) of the Forward and Reverse UOT alignments and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-OR Estimates word correspondences using the union (OR) of the Forward and Reverse UOT alignments and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-FORWARD Estimates word correspondences using the Forward (source→target) UOT alignment and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-REVERSE Estimates word correspondences using the Reverse (target→source) UOT alignment and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

BASELINE Uses no word correspondences and predicts translation errors using an LLM based only on the source – target text.

See Section 2.2, Section 2.2, and Section 2.3 for details. The correspondence between the submitted result folder names and the system names used in this paper is shown in Table 1.

Setting	Detection F1			Recall (gold category)			Ann/claim	Sev. agr.	Pearson r	Spearman ρ
	all	output	span	Acc.	Term.	Flu.				
LLM	**0.512**	**0.555**	**0.283**	0.40	0.34	0.51	11.2	0.47	0.65	0.63
UOT-AND	0.486	0.528	0.254	0.36	0.17	0.48	11.2	0.46	0.64	0.59
UOT-REVERSE	0.485	0.523	0.276	0.35	0.17	0.51	10.3	0.47	0.60	0.65
UOT-OR	0.482	0.526	0.261	0.35	0.34	0.52	10.6	0.42	0.61	0.59
BASELINE	0.475	0.510	0.255	0.34	0.17	0.51	11.1	0.45	0.63	0.62
UOT-FORWARD	0.454	0.485	0.256	0.28	0.31	0.52	11.0	0.39	0.63	0.56

Table 2: Official results for the Ehime-U submissions to the WAT 2026 Japanese–English Patent Claims Translation/Evaluation Task. Detection F1 measures error-detection performance against the gold reference. The *all* setting counts errors on both the input and output sides, whereas *output* counts only errors on the output (target) side. The *span* score is span-level detection F1 at an intersection over union (IoU) threshold of 0.5. Recall is calculated over gold-reference characters belonging to each MQM category: Acc. denotes accuracy, Term. denotes terminology consistency, and Flu. denotes fluency. Ann/claim is the average number of submitted annotations per claim. Sev. agr. is the severity agreement rate over characters marked as errors by both the gold reference and the submission. Pearson r and Spearman ρ measure claim-level correlations between the submitted quality scores and the gold MQM scores, respectively. The gold annotation density is 9.3 annotations per claim for JA–EN. Correlations should be interpreted with caution because they are calculated over only 10 claims.

Setting	Aligned src	Null src	Aligned tgt	Null tgt
LLM	3113	1232	2667	399
UOT-AND	1346	2999	1172	1894
UOT-OR	1508	2837	1192	1874
UOT-FORWARD	1508	2837	1190	1876
UOT-REVERSE	1174	1892	1346	1720
BASELINE	—	—	—	—

Table 3: Correspondence coverage per setting. "Aligned" counts tokens with at least one link, and "Null" counts tokens with no link. For UOT-REVERSE, the source columns count English tokens and the target columns count Japanese tokens, so its totals are not directly comparable to the other rows.

4 Experimental Results

4.1 Effect of Word Correspondence on Error Prediction

We compare the six settings using the official scores provided in the task organizers' result report (Nakazawa et al., 2026).

Table 2 summarizes the official results for all six settings, including character-level and span-level error-detection F1, category-specific recall, annotation density, severity agreement, and claim-level score correlations. We count a reference error as detected if a predicted error span overlaps with it; a predicted error that overlaps with no reference error is counted as a false positive.

For character-level detection F1 on both the input and output sides (the *all* metric), the four UOT variants achieve an average score of 0.477. The best UOT setting, UOT-AND, reaches 0.486, outperforming BASELINE (0.475). LLM achieves the highest score (0.512), suggesting that the LLM-based alignment is the most effective of the six settings under this metric. Among the UOT variants, UOT-AND (0.486) and UOT-REVERSE (0.485) perform best, whereas UOT-FORWARD (0.454) performs worst. These results suggest that the choice of alignment direction and combination strategy affects error-prediction performance.

The category-specific recall results reveal differences across error categories. Compared with BASELINE, LLM improves recall for accuracy from 0.34 to 0.40 and for terminology consistency from 0.17 to 0.34, while fluency recall remains unchanged at 0.51. This pattern suggests that word-correspondence information may be particularly helpful for identifying accuracy- and terminology-related errors.

4.2 Correspondence Statistics

Table 3 reports correspondence coverage on both the source and target sides. "Aligned src" counts source tokens with at least one link, and "Null src" counts source tokens with no link; "Aligned tgt" and "Null tgt" are defined analogously for target tokens.

Reverse is shown from the target side, so its two "source" columns count English tokens and its two "target" columns count Japanese tokens; its totals are not directly comparable to the other rows.

The statistics show that, under the hyperparam-

Error label	Main category	Subcategory	Sub-subcategory
訳抜け (omission)	Accuracy	Omission	–
未翻訳 (untranslated text)	Accuracy	Untranslated text	–
湧き出し (addition)	Accuracy	Addition	–
誤訳 (mistranslation)	Accuracy	Mistranslation	–
(誤訳) 数値・記号 (numerals/symbols)	Accuracy	Mistranslation	Numerals/Symbols
(誤訳) 冠詞 (article)	Accuracy	Mistranslation	Article
(誤訳) 係り先誤り (incorrect dependency)	Accuracy	Mistranslation	Incorrect dependency
(誤訳) 係り先が不明 (unknown dependency)	Accuracy	Mistranslation	Unknown dependency
文脈上不適切 (訳語選択)	Terminology	Inappropriate for context	–
句読点 (punctuation)	Fluency	Punctuation	–
文法 (grammar)	Fluency	Grammar	–
ぎこちない (awkward)	Style	Awkward	–
用語の一貫性 (inconsistent terminology)	Terminology	Inconsistent use	–

Table 4: Mapping of the error labels used in our experiments to the MQM taxonomy.

Setting	Top error types
LLM	訳抜け (41), 誤訳 (25), 冠詞 (19)
UOT-AND	訳抜け (40), 誤訳 (26), 冠詞 (19)
UOT-OR	訳抜け (39), 誤訳 (25), 冠詞 (16)
UOT-FORWARD	訳抜け (39), 誤訳 (24), 冠詞 (16)
UOT-REVERSE	訳抜け (39), 誤訳 (24), 冠詞 (18)
BASELINE	訳抜け (39), 誤訳 (30), 冠詞 (18)

Table 5: Top three error types per setting.

Setting	Total	Major	Minor
GOLD	65	40	25
LLM	112	37	75
UOT-AND	112	41	71
UOT-OR	106	36	70
UOT-FORWARD	110	37	73
UOT-REVERSE	103	39	64
BASELINE	111	34	77

Table 6: Error counts per setting. The GOLD row reports the number of errors annotated in the gold reference.

eter settings used in our experiments, LLM predicted substantially more correspondences than any UOT-based setting. On the source side, LLM links 3,113 tokens, whereas the UOT variants link 1,174 – 1,508; on the target side, LLM links 2,667 tokens, compared with 1,172 – 1,346 for UOT variants. Under these experimental conditions, the LLM-based alignment is considerably denser than the UOT alignments.

4.3 Error Type Distribution

Table 4 presents the mapping between all error labels used in our experiments and the corresponding categories in the MQM taxonomy. Table 5 lists the three most frequent error labels for each setting. The same three labels were the most frequent across all settings: 訳抜け (omission), 誤訳 (mistranslation), and 冠詞 (article error). 訳抜け was the most frequent overall, appearing 39–41 times per setting, indicating that omission was the dominant error type in this dataset regardless of the alignment setting.

Table 6 compares the number and severity of errors in the gold reference with those predicted by the six settings. The gold reference contains 65 annotated errors, including 40 major and 25 minor errors. In contrast, the six settings predicted between 103 and 112 errors in total, substantially more than the number of errors annotated in the gold reference.

Among the six settings, UOT-AND and LLM predicted the most errors (112 each), whereas UOT-REVERSE predicted the fewest (103). The number of predicted major errors ranged from 34 to 41, with UOT-AND predicting the most (41) and BASELINE the fewest (34). The number of predicted minor errors ranged from 64 to 77, with BASELINE predicting the most (77) and UOT-REVERSE the fewest (64).

Overall, under the hyperparameter settings used in this study, all six settings predicted more errors than were annotated in the gold reference. The predicted error counts and severity distributions also varied across settings, although the three most frequent error labels remained the same. The differences between settings are small overall (Table 5).

4.4 Qualitative Observations

We observe that omission (訳抜け) and mistranslation (誤訳) errors dominate in every setting, and that the three most frequent error types are the same across all six settings. This suggests that the choice of word correspondence mainly affects the number of errors the model reports, not the set of dominant error types. The differences between settings are small: UOT-AND predicted the most major errors,

while BASELINE predicted the fewest major but the most minor errors. The latter is consistent with the no-alignment baseline flagging more low-severity issues when it cannot rely on correspondence information.

5 Conclusion

We presented a three-stage error prediction system for patent claim translations that used word-level correspondence as auxiliary information. The first stage tokenized the source and target claims using language-specific tokenizers. The second stage predicted word correspondences using five approaches: four unbalanced optimal transport (UOT) alignment strategies (Forward, Reverse, AND, and OR) and one LLM-based alignment method. Together with a no-alignment baseline, these approaches yielded six experimental settings. The third stage prompted an LLM to predict MQM errors using the word correspondences, or without them in the baseline setting, along with one few-shot example and a filtered correspondence list.

Among the six settings, the LLM-based alignment achieved the highest error-detection F1. It also outperformed the no-alignment baseline (0.512 vs. 0.475; Table 2). Compared with the gold reference annotations, the improvement varied across error categories: recall for accuracy and terminology consistency increased, whereas recall for fluency errors remained unchanged (Table 2). The three most frequent error types were the same across all settings (訳抜け, 誤訳, and 冠詞; Table 5), while the total number of predicted errors varied only slightly (103–112; Table 6). These results suggest that word correspondence can improve error-detection performance, particularly for accuracy- and terminology-related errors, although the observed performance differences were relatively small and depended on the experimental settings.

Future work includes tuning the unbalanced optimal transport parameters (ε, τ, link threshold, and row mass floor) on development data, and extending the evaluation to more translation directions.

Acknowledgments

This work was supported by joint research funding from Mirai Translate, Inc.

References

Yuki Arase et al. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kosei Buma, Takehito Utsuro, and Masaaki Nagata. 2025. Two step automatic post editing of patent machine translation based on pre-trained encoder models and LLMs. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 218–231, Mumbai, India. Association for Computational Linguistics.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, pages 2292–2300.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*.

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

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303.

Matthias Liero, Jürgen Jost, and Giandomenico Mielke. 2018. Optimal transport in competition with reaction: The hellinger–kantorovich distance. *SIAM Journal on Mathematical Analysis*, 50(1):1170–1204.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2013. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. pages 455–463.

Qingyu Lu et al. 2024. Error analysis prompting enables human-like translation evaluation in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Taiki Ushimaru, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2026. Findings of the 2nd patent claims translation/evaluation task. In *Proceedings of the 13th Workshop on Asian Translation*.

Toshiaki Nakazawa et al. 2025. Findings of the first patent claims translation task at WAT2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*.

A Prompts

This appendix lists the prompts used in the pipeline. All prompts were issued to GPT-6-Astra through an OpenAI API with reasoning effort `low` and temperature 0. Placeholders in curly braces are replaced at run time.

A.1 LLM-based Token-Pair Scoring

The UOT score stage asks the model to score how strongly each source token corresponds to each target token, so that a downstream UOT solver can find the globally optimal word alignment.

```
[System]
You are an expert bilingual patent linguist.

SOURCE LANGUAGE: {source_language}
TARGET LANGUAGE: {target_language}

You are given the tokenized SOURCE and TARGET
    as LOCAL token IDs.
Your job is NOT to produce the final alignment.
      Your job is to score how
strongly each SOURCE token corresponds to each
     TARGET token, so that a
downstream Unbalanced Optimal Transport (UOT)
    solver can find the globally
optimal word alignment.

For every SOURCE token that has a plausible
    translation in the TARGET, list
the candidate TARGET tokens and a score in [0,
     1]:

- 1.0  = this is the exact translation of the
    source token
- 0.7  = a strong candidate (synonym,
    inflection, or the same term)
- 0.4  = a weak candidate (related meaning,
    part of the same phrase)
- omit = not a plausible counterpart

Rules:
- Score by MEANING, not surface form.
    Japanese and English rarely share
  characters, so a lexical match is usually
    meaningless.
- Repeated terms (前記 / "the", 出力電圧 / "
    output voltage") should be scored
  for EVERY occurrence, by syntactic position.
- Function words (の, を, に, が, は, と)
    usually map to an English
  preposition/determiner or to nothing; score
    them 0.4-0.7 where a target
  counterpart exists.
- List at most 5 target candidates per source
    token.
- You do NOT need to cover every source token;
     the solver handles gaps.

Return JSON only:
{"scores": [{"source": 0, "target": 3, "score":
     0.9},
            {"source": 1, "target": 4, "score":
     0.6}]}

[User]
SOURCE TOKENS:
{src_listing}

TARGET TOKENS:
{tgt_listing}

Score the plausible source->target token
    correspondences.
- List only pairs that are plausible
    counterparts.
- Use a score in [0, 1].
- You do NOT need to cover every source token.

Return JSON only.
```

A.2 Word Alignment

The alignment stage asks the model to produce span boundaries and a token-level alignment in one JSON object.

```
[System]
You are an expert bilingual patent linguist.

SOURCE LANGUAGE: {source_language}
TARGET LANGUAGE: {target_language}

You are given the tokenized SOURCE and TARGET
    as LOCAL token IDs.
Do TWO things in ONE answer:

(A) SPAN CONSTRUCTION -- mark, for EACH side
    independently, where each
new span begins. Output "source_breaks" / "
    target_breaks": the token
index where each new span begins. Index 0
    always starts the first span.

(B) TOKEN-LEVEL ALIGNMENT -- state which
    SOURCE TOKENS correspond to
which TARGET TOKENS. Use the DIRECT token IDs
    you were shown
([S0], [S1], ... / [T0], [T1], ...), NOT span
```

```
    ordinals.

CRITICAL RULES FOR ALIGNMENT:
1. Each alignment entry has EXACTLY ONE source
     token in "source_tokens".
2. A source token may map to ZERO, ONE, or TWO
     target tokens.
3. ALIGN LEXICAL MEANING, not phrase meaning.
4. For noun phrases, align nouns to nouns and
    modifiers to modifiers.
5. Repeated words must be resolved
    independently.
6. Match repeated terms by SYNTACTIC POSITION,
     not surface proximity.
7. A target token should normally be the
    counterpart of AT MOST ONE
source token.
8. Word order may differ.
9. Leave a source token unaligned ([]) ONLY
    when it is a Japanese
grammatical particle.

Return JSON only:
{"source_breaks": [0, 2, 3], "target_breaks":
    [0, 3],
 "alignments": [{"source_tokens": [0], "
    target_tokens": [1]}, ...]}

[User]
SOURCE TOKENS:
{src_listing}

TARGET TOKENS:
{tgt_listing}

Mark the span boundaries for each side, then
    align each SOURCE token to
its direct lexical translation equivalent in
    the TARGET.
- ONE source token per alignment entry.
- ZERO, ONE, or TWO target tokens. Never more
    than TWO.
- If there is no direct lexical counterpart,
    return [].
- Search the entire target window; do not rely
     on token position.

Return JSON only.
```

A.3 Error Prediction

The error prediction stage asks the model to report translation errors in the MQM schema. The prompt contains the source and target claims, the word correspondence list, the token ID lists, the MQM error type definitions, and one few-shot example.

```
[System]
You are an expert MQM patent translation
    annotator. Output exactly one
JSON object, nothing else.

[User]
Report all translation errors you can identify.

Perform the analysis in multiple passes:
1. Check every source content word and short
    phrase against its
translation.
2. Check every target content word and short
    phrase for unsupported
additions.
3. Check numbers, symbols, terminology,
    articles, grammar, dependencies,
and punctuation.
4. Check whether any source content is missing
     or left untranslated.
5. Check consistency across the whole
    translation.
6. After completing all checks, output every
    independently identifiable
error.

Do not stop when you find a few errors.
Do not assume that a generally understandable
    translation is error-free.

SOURCE ({source_language}):
{source_text}

TARGET ({target_language}):
{target_text}

TOKEN IDS (use these IDs for every error span):

{token_listing}

{alignment_section}

Output shape (exactly this JSON, no prose, no
    markdown):
{"score": 85, "errors": [{"side": "output",
 "target_token_ids": ["T0"], "text": "<
    verbatim span>",
 "error": "<label>", "severity": "小"}]}

The "score" must be an integer from 0 to 100.
    Do NOT output
"start"/"end"; Python computes offsets. For
    output errors return
target_token_ids; for input errors return
    source_token_ids. Use the
smallest consecutive token ID span covering
    the error.

湧き出し = Translation includes information
    that is not present in the source and that
     is not supposed to be included
訳抜け = Translation is missing content from
    the source and the omission is
    inappropriate.
未翻訳 = Source text has been left untranslated
誤訳 = Translaiton does not accurately
    represent the source.
(誤訳)数値・記号 = Translation errors related
    to numerals and symbols.
(誤訳)冠詞 = Incorrect use of articles
(誤訳)係り先誤り = The adjective phrase or
    parallel structure has an incorrect
    dependency (please point out the correct
    dependency)
(誤訳)係り先が不明 = The dependency structure
    of the source is not maintained.
(誤訳)解釈の曖昧性 = The translation is more
    ambiguous than the source text (e.g. the
    source text can be interpreted in two ways
```

```
    , whereas the translation can be
    interpreted in three or more ways).
句読点 = Incorrect punctuation (for locale or
    style, including improper sentence
    division, since patent claims must be
    written in one sentence). 1) An English
    text uses a semicolon where a comma should
     be used. 2) A two-digit year reference
    begins with an open single quote instead
    of a close single quote (apostrophe). 3) A
     Greek text uses a question mark instead
    of the anticipated semicolon to express a
    question. 4) German quotation marks are
    carried over into English or French target
     content.
スペリング = Incorrect spelling or
    capitalization.
文法 = Problems with grammar, other than
    orthography.
言語使用域 = Wrong grammatical register (eg,
    inappropriately informal pronouns).
一貫性の欠如 = Internal inconsistency (not
    related to terminology)
文字エンコーディング = Characters are garbled
    due to incorrect encoding.
文脈上不適切 (訳語選択) = Terminology is non-
    standard or does not fit context.
用語の一貫性 = Terminology is used
    inconsistently.
ぎこちない = Translation has stylistic problems
    .
その他 = Any other issue.
原文エラー = An error in the source.
非翻訳 = Impossible to reliably characterize
    distinct errors.

SPAN RULE (very important): for each error,
    quote the MINIMAL sub-span
that IS the error -- ONE word or ONE short
    phrase. Do NOT quote the
surrounding clause.

Identify all errors you can find.
Example (format only):
{fewshot}

Return exactly one JSON object.
```

A.3.1 No-Alignment Baseline

The error prediction stage asks the model to report translation errors in the MQM schema. The prompt contains the source and target claims, the token ID lists, the MQM error type definitions, and one few-shot example.

```
\begin{lstlisting}
[System]
You are an expert MQM patent translation
    annotator. Output exactly one
JSON object, nothing else.

[User]
Report all translation errors you can identify.

Perform the analysis in multiple passes:
1. Check every source content word and short
    phrase against its
   translation.
2. Check every target content word and short
    phrase for unsupported
   additions.
3. Check numbers, symbols, terminology,
    articles, grammar, dependencies,
   and punctuation.
4. Check whether any source content is missing
     or left untranslated.
5. Check consistency across the whole
    translation.
6. After completing all checks, output every
    independently identifiable
   error.

Do not stop when you find a few errors.
Do not assume that a generally understandable
    translation is error-free.

SOURCE ({source_language}):
{source_text}

TARGET ({target_language}):
{target_text}

TOKEN IDS (use these IDs for every error span):

{token_listing}

Output shape (exactly this JSON, no prose, no
    markdown):
{"score": 85, "errors": [{"side": "output",
 "target_token_ids": ["T0"], "text": "<
    verbatim span>",
 "error": "<label>", "severity": "小"}]}

The "score" must be an integer from 0 to 100.
    Do NOT output
"start"/"end"; Python computes offsets. For
    output errors return
target_token_ids; for input errors return
    source_token_ids. Use the
smallest consecutive token ID span covering
    the error.

湧き出し = Translation includes information
    that is not present in the source and that
     is not supposed to be included
訳抜け = Translation is missing content from
    the source and the omission is
    inappropriate.
未翻訳 = Source text has been left untranslated
誤訳 = Translaiton does not accurately
    represent the source.
(誤訳)数値・記号 = Translation errors related
    to numerals and symbols.
(誤訳)冠詞 = Incorrect use of articles
(誤訳)係り先誤り = The adjective phrase or
    parallel structure has an incorrect
    dependency (please point out the correct
    dependency)
(誤訳)係り先が不明 = The dependency structure
    of the source is not maintained.
(誤訳)解釈の曖昧性 = The translation is more
    ambiguous than the source text (e.g. the
    source text can be interpreted in two ways
    , whereas the translation can be
    interpreted in three or more ways).
句読点 = Incorrect punctuation (for locale or
    style, including improper sentence
```

```
    division, since patent claims must be
    written in one sentence). 1) An English
    text uses a semicolon where a comma should
     be used. 2) A two-digit year reference
    begins with an open single quote instead
    of a close single quote (apostrophe). 3) A
     Greek text uses a question mark instead
    of the anticipated semicolon to express a
    question. 4) German quotation marks are
    carried over into English or French target
     content.
スペリング = Incorrect spelling or
    capitalization.
文法 = Problems with grammar, other than
    orthography.
言語使用域 = Wrong grammatical register (eg,
    inappropriately informal pronouns).
一貫性の欠如 = Internal inconsistency (not
    related to terminology)
文字エンコーディング = Characters are garbled
    due to incorrect encoding.
文脈上不適切 (訳語選択) = Terminology is non-
    standard or does not fit context.
用語の一貫性 = Terminology is used
    inconsistently.
ぎこちない = Translation has stylistic problems
    .
その他 = Any other issue.
原文エラー = An error in the source.
非翻訳 = Impossible to reliably characterize
    distinct errors.

SPAN RULE (very important): for each error,
    quote the MINIMAL sub-span
that IS the error -- ONE word or ONE short
    phrase. Do NOT quote the
surrounding clause.

Identify all errors you can find.
Example (format only):
{fewshot}

Return exactly one JSON object.
```

A.4 Few-Shot Example Format

The few-shot example is a gold-annotated block rendered in the same format as the expected output:

```
SOURCE: {truncated source claim}
TARGET: {truncated target claim}
ERRORS: [{"side": ..., "text": ..., "error":
    ..., "severity": ...}, ...]
```