# Plug, Play, and Fuse: Zero-Shot Joint Decoding via Word-Level Re-ranking Across Diverse Vocabularies

**Sai Koneru**[1], **Matthias Huck**[2], **Miriam Exel**[2], and **Jan Niehues**[1]
[1] Karlsruhe Institute of Technology
[2] SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
{sai.koneru, jan.niehues}@kit.edu
{matthias.huck, miriam.exel}@sap.com

## Abstract

Recent advancements in NLP have resulted in models with specialized strengths, such as processing multimodal inputs or excelling in specific domains. However, real-world tasks, like multimodal translation, often require a combination of these strengths, such as handling both translation and image processing. While individual translation and vision models are powerful, they typically lack the ability to perform both tasks in a single system. Combining these models poses challenges, particularly due to differences in their vocabularies, which limit the effectiveness of traditional ensemble methods to post-generation techniques like N-best list re-ranking. In this work, we propose a novel zero-shot ensembling strategy that allows for the integration of different models during the decoding phase without the need for additional training. Our approach re-ranks beams during decoding by combining scores at the word level, using heuristics to predict when a word is completed. We demonstrate the effectiveness of this method in machine translation scenarios, showing that it enables the generation of translations that are both speech- and image-aware while also improving overall translation quality[1].

## 1 Introduction

A broad spectrum of Large Language Models (LLMs) are being developed at an increasing pace, with efforts focused alone or together on adapting them to specific domains (Roziere et al., 2023; Bolton et al., 2024; Colombo et al., 2024), enhancing their ability to process multiple modalities (Liu et al., 2023; Tang et al., 2023; Li et al., 2024; Beyer et al., 2024), or training general-purpose LLMs using high-quality data, advanced architectures, and

larger numbers of parameters (Touvron et al., 2023; Dubey et al., 2024; Jiang et al., 2023a; Mesnard et al., 2024). As a result, numerous models are now publicly available, each with its own unique strengths and weaknesses.

Many use cases, such as image-aware translation in movie subtitling, require combining these strengths because visual cues can be essential for disambiguating the text and ensuring accurate translations.. Currently, LLMs, such as Tower (Alves et al., 2024), Alma-R (Xu et al., 2024a), and Madlad-400 (Kudugunta et al., 2024), excel at translation tasks (Kocmi et al., 2024), while models like PaliGemma (Beyer et al., 2024) and LLava (Li et al., 2024) are leading in vision-related tasks. To effectively address image-aware translation, it is essential to harness the strengths of both translation and vision models.

One way to address such a task is to train a multimodal LLM to enhance its translation capabilities without compromising its vision abilities or vice versa. However, this approach requires additional training and task-specific data. Another approach is to leverage ensembling the two models via shallow fusion (Gulcehre et al., 2015) or re-ranking the N-best list (Hasan et al., 2007). The disadvantage of shallow fusion is that it assumes both models share the same vocabulary, which is often not the case with current open-source models.

Additionally, re-ranking the N-best list is insufficient because it doesn't allow models to influence each other during decoding. For example, in Figure 1, translating from English to gender-marked language French using audio and transcript shows this limitation. The Speech Translation (ST) model correctly uses the speaker's voice to translate "fell" into the right gender form but misidentifies the name "Ples." On the other hand, the Machine Trans-

---

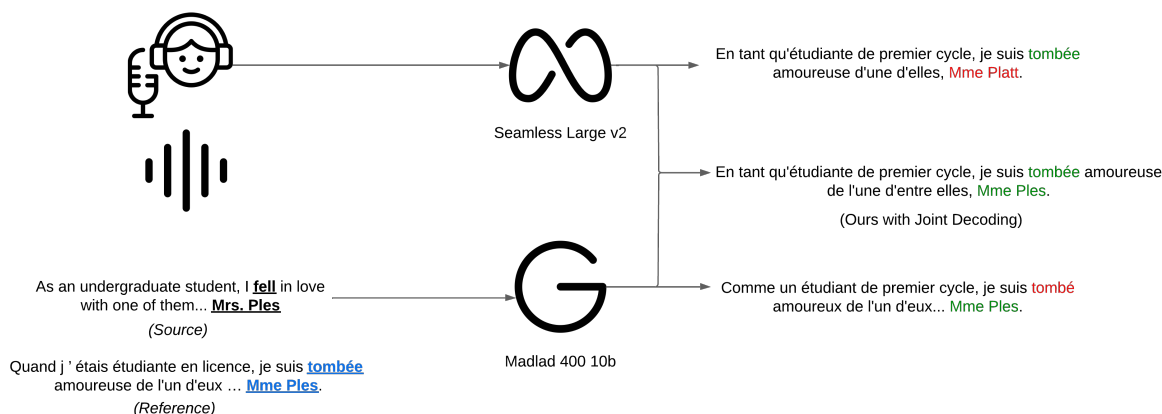[1] Code can be found at: https://ai4lt.anthropomatik.kit.edu/english/projects_kontextmt.php

Figure 1: The source sentence to be translated is ambiguous because the translation of the word **"fell"** can be either masculine **("tombé")** or feminine **("tombée")**, depending on the speaker's gender. Seamless-Large V2 (Barrault et al., 2023) utilizes audio cues to correctly determine the gender form but struggles to accurately translate the name **"Mrs Ples"** using audio alone. In contrast, the text translation model Madlad-400-10b-mt (Kudugunta et al., 2024) relies on the gold transcript to correctly translate the name but fails to resolve the gender ambiguity. By combining both models using our approach, the translation correctly captures both the gender form and the named entity.

lation (MT) model correctly translates the name but can't use the speaker's voice for gender disambiguation. Thus, re-ranking falls short, as the correct forms may not even be in the N-best list due to low probability with missing cues.

Furthermore, re-ranking during the decoding process is impractical because the hypotheses are partial and may not align with the tokenization of the ranker model, leading to incorrect probability estimates (Section 2.1). Thus, resolving vocabulary mismatches by mapping the vocabulary of one model to another (Minixhofer et al., 2024; Xu et al., 2024b) is necessary to allow the merging of probabilities during decoding. However, this approach requires significant additional training steps and can lead to deviations from the original model. Therefore, developing a plug-and-play approach that seamlessly combines different models without requiring additional training or task-specific data is highly advantageous.

This work aims to enable the ranker model to influence the decoding process (online) without any constraints compared to conventional offline N-best list re-ranking. We address this by ensuring that the ranker model only influences the scores for completed words and not for the last word if it is unfinished. Additionally, we propose using the ranker model to determine whether the last word is finished rather than relying on look-ahead

approaches to maintain efficiency.

Our main contributions are summarized below:

1. **Online Re-Ranking Algorithm**: We introduce a novel re-ranking algorithm that operates at the word level during decoding at sub-word level, allowing for more accurate tokenization and better integration of information from different models

2. **Plug-and-Play Approach**: Our method does not require additional training or task-specific data, making it a flexible and practical solution for integrating multiple models with different strengths.

3. **Context-aware Translations**: We demonstrate through experiments including targeted multimodal test sets, which require information from both modalities, that our approach effectively combines the strengths of different models and improves translation quality (Illustrated in Figure 1).

## 2 Methodology

Given that many models are trained on different tasks, architectures, modalities, and data types, combining these models to leverage inputs from multiple modalities and facilitate knowledge sharing is highly beneficial. Moreover, it is ideal if the

ensembling approaches satisfy the following constraints: 1) It should not rely on shared vocabularies for flexibility in choosing models and maximizing potential combinations. 2) Effective knowledge sharing should occur during decoding to better navigate the search space exploiting this knowledge at each step. 3) Avoid requiring additional training, parameters, or major dependence on task-specific data for maximum applicability and not cause deviations from the pre-trained model.

This section presents our algorithm for ensembling models with different vocabularies that satisfy the aforementioned constraints. First, we explain why re-ranking partial hypotheses can lead to incorrect probability estimates if the word is incomplete. Next, we introduce and justify a heuristic-based approach that predicts whether a hypothesis is at the end of a word, allowing for accurate re-ranking of completed words in partial hypotheses. Finally, we formally describe the complete algorithm, detailing how we merge probabilities from different models and how this process can be integrated with decoding strategies.

## 2.1 Challenges of Re-Ranking Partial Hypotheses

Current Neural Machine Translation (NMT) and LLM-based models can utilize various tokenization methods, such as byte-pair encoding (BPE) (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018). These methods often result in distinct vocabularies due to variations in the data and tokenizer training processes. Despite these differences, techniques like re-ranking can still enable estimating the probability of sentences generated from another model. This is achieved by detokenizing the hypothesis from generator model and re-tokenizing it using the ranker model's vocabulary. This process enables the ranker model to produce accurate probability estimates based on its own tokenization scheme.

Now, consider the case of re-ranking while the hypotheses are still being decoded. Assume we have model $\mathcal{M}_G$ (the generator) and model $\mathcal{M}_R$ (the ranker), each using different tokenizers assign all the tokens in the sentence "Decoding is awesome" with a probability of p for a particular input. However, $\mathcal{M}_G$ tokenizes the sentence with sub-word tokens as "Dec od ing _is _awe some," while

$\mathcal{M}_R$ would tokenize it as "Dec od ing _is _awes ome."

If we attempt to re-rank during the decoding process, $\mathcal{M}_R$ will provide correct probability estimates up until "_is" is generated. However, when the generator predicts "_awe," $\mathcal{M}_R$ would incorrectly estimate the probability because it expects "_awes" instead. Even though both models aim to generate the same sentence, this tokenization mismatch leads to incorrect probability estimates during the decoding process, making online re-ranking challenging.

## 2.2 End-of-Word Prediction in Decoding for Accurate Re-Ranking

While the partially generated hypothesis cannot be accurately ranked at every time step, consider the cases when each word is finished. At that time, we can re-rank the complete hypothesis as the last word is fully generated and the ranker model can tokenize the completed word as it would have done naturally, thereby providing accurate probability estimates. If we know that the last word is incomplete, we can use this information to wait and only rank the previously completed words. Knowing the end of the word enables more precise re-ranking during decoding, even with models that use different tokenization schemes.

Nonetheless, a significant challenge remains: how do we determine when the last word is completed? If the tokenizer places spaces at the right of characters, we could check the predicted token to see if it includes a space, signaling the end of a word. However, this approach is not universal, as many tokenizers do not follow this pattern, and we aim to develop a tokenizer-agnostic solution.

One alternative is to perform a look-ahead step to check if the word has been completed, but this method is also sub-optimal, as it would require decoding twice for each step in the generation process, significantly increasing computational complexity and reducing efficiency. We need a more efficient and generalizable method to determine when a word has been completed during decoding.

To address these challenges, we propose using the ranker model to predict the next token and determine if the word has been completed. This approach offers two key advantages.

Firstly, if the ranker model predicts a space as

---

**Algorithm 1** Computing merged score of candidate with generator and ranker models.

1: **procedure** MERGESCORE
2:    **Input:** Generator tokens $g_1, g_2, g_3, \ldots, g_n$, Reranker tokens $r_1, r_2, r_3, \ldots, r_m$, Generator Model $\mathcal{M}_G$, Ranker model $\mathcal{M}_R$, Generator Input $\mathcal{I}_G$, Ranker Input $\mathcal{I}_R$, Re-ranking weight $\alpha$,
3:    **Output:** $merged\_score$
4:    $next\_tok \leftarrow \arg\max \log \mathcal{P}(y|r_1, \ldots, r_m; \mathcal{I}_R; \mathcal{M}_R)$
5:    **if** $next\_tok[0] ==$ "_" or $next\_tok ==$ "&lt;eos&gt;" **then**
6:        $full_G \leftarrow \frac{1}{n} \sum \log \mathcal{P}(g_1, g_2, \ldots, g_n|\mathcal{I}_G; \mathcal{M}_G)$                    ▷ Generator Score for all words
7:        $full_R \leftarrow \frac{1}{m} \sum \log \mathcal{P}(r_1, r_2, \ldots, r_m|\mathcal{I}_R; \mathcal{M}_R)$                    ▷ Ranker Score for all words
8:        $merged\_score \leftarrow (\alpha) \times full_G + (1 - \alpha) \times full_R$
9:    **else**
10:        $[g_1, \ldots, g_j], [g_{j+1}, \ldots, g_n] \leftarrow \text{split\_candidate}(g_1, \ldots, g_n)$          ▷ Last word from j+1 token
11:        $[r_1, \ldots, r_k], [r_{k+1}, \ldots, r_m] \leftarrow \text{split\_candidate}(r_1, \ldots, r_m)$          ▷ Last word from k+1 token
12:        $prev_G \leftarrow \frac{1}{j} \sum \log \mathcal{P}(g_1, g_2, \ldots, g_j|\mathcal{I}_G; \mathcal{M}_G)$          ▷ Generator Score for previous words
13:        $prev_R \leftarrow \frac{1}{k} \sum \log \mathcal{P}(r_1, r_2, \ldots, r_k|\mathcal{I}_R; \mathcal{M}_R)$          ▷ Ranker Score for previous words
14:        $prev_{GR} \leftarrow (\alpha) \times prev_G + (1 - \alpha) \times prev_R$
15:        $last_G \leftarrow \sum \log \mathcal{P}(g_{j+1}, \ldots, g_n|\mathcal{I}_G; \mathcal{M}_G)$
16:        $merged\_score \leftarrow \frac{1}{n}[prev_{GR} \times j + last_G]$                    ▷ Re-normalized merged score
17:    **end if**
18: **end procedure**

---

the next top character, it indicates that the current last word has been completed. The hypothesis will be tokenized correctly, given that it is the prediction from the ranker model itself. Secondly, this prediction can be done together with the re-ranking process by simply also predicting the next token given the previous tokens of the current hypothesis to the ranker model.

This method is more efficient than the look-ahead approach, requiring only one pass of the generator and the ranker model. In contrast, the look-ahead method would require two passes of the generator and one pass of the ranker model. Using the ranker model in this way, we can ensure proper tokenization and accurate probability estimates during the decoding process (online) without additional computational overhead.

## 2.3 Integrating Online Re-Ranking with Search

This section formalizes achieving online re-ranking at a word level using beam search as an example of a decoding strategy. Note that the approach can also be applied to other strategies, with slight modifications when necessary.

A set of candidate sequences is typically maintained during the search, with the number of candidates equal to the configured beam size $b$. At each time step, for each of the $b$ candidate sequences, the model computes likelihood scores for all possible token extensions based on the vocabulary size $V$. This results in a total of $b \times V$ possible extensions. From these $b \times V$ extensions, the top $b$ sequences with the highest scores are selected to form the new set of candidate sequences. This process is repeated iteratively, updating the candidate sequences at each step until enough beams are generated that include end-of-sentence tokens or until a predefined length limit is reached.

To enable re-ranking during the decoding process, we need to adjust the scores of the possible extensions using the ranker model. Directly calculating the likelihood of all extensions would be computationally impractical. Therefore, we introduce a new parameter $topk$, which selects the top $topk$ extensions for each beam during re-ranking.

Hence, at each time step, the generator model calculates the likelihood scores for all $V$ possible extensions for each of the $b$ candidate sequences, resulting in $b \times V$ extensions. Instead of re-ranking all $b \times V$ extensions, the top $topk$ extensions with the highest likelihood scores are selected for each beam. Thus, only $b \times topk$ extensions are considered during re-ranking. For the selected $b \times topk$ extensions, the ranker model estimates their scores

and combines them with the original generator scores. For the remaining $b \times (V - topk)$ extensions, the scores are set to $-\infty$ (logically equivalent to discarding them) since they would not be selected in the top beams.

This method significantly reduces computational complexity while allowing effective re-ranking of the most promising candidate extensions, improving the decoding process.

At every decoding step, the problem can be reformulated as determining the merged score of the top candidates according to both models.

When calculating the merged score during decoding, it's essential to exclude the ranker model's probability if the last word in the current beam is incomplete. This prevents incomplete words from skewing the final score. For beams with incomplete final words, we combine the joint scores of the preceding words with the generator's score for the last word, ensuring proper normalization to address scale differences between finished and unfinished beams.

After computing the merged scores, we select the top extensions and repeat the process until all beams reach the end-of-sentence token. This method ensures that the final translation is based on fully formed words, optimizing the ranker model's effectiveness and maintaining consistent scoring across all candidates.

### 2.3.1 Unified Scoring with Generator and Ranker

The algorithm to compute the merged score is formally defined in Algorithm 1 and explained below.

Let us consider two models: the Generator $\mathcal{M}_G$ and the Ranker $\mathcal{M}_R$. Let $\mathcal{C}$ denote the current candidate for re-ranking and inputs $\mathcal{I}_G$ and $\mathcal{I}_R$ for $\mathcal{M}_G$ and $\mathcal{M}_R$ respectively.

Let the full candidate $\mathcal{C}$ consist of tokens $g_1, g_2, g_3, \ldots, g_n$ and $r_1, r_2, r_3, \ldots, r_m$ according to $\mathcal{M}_G$ and $\mathcal{M}_R$, respectively. Note that $n$ and $m$ denote the length of the sequence, and they may differ due to different tokenization.

The key idea is to rank and merge scores for completed words. We use the ranker model to predict the next token and determine if the last word is finished (**Line 4**).

**If the last word is finished**: We can calculate the probability of the full sequence in this case, sim-

ilar to the case of N-best list re-ranking. First, we calculate the likelihood of the candidate by averaging the log probabilities for both the generator and the ranker (**Line 6-7**). Then, we merge the scores from both models to determine the final score for the candidate sequence using a hyper-parameter $\alpha$ for weighting (**Line 8**). This combined score considers the estimates from both models, allowing for contributions from both models.

**If the last word is incomplete**: We cannot rank the last word due to potential incorrect tokenization. However, we can still estimate the tokens preceding the last word using the ranker model and merge their probabilities. First, we split the candidate into previous and last words based on the ranker and generator (**Lines 10-11**). We compute the merged score for the previous words using the weighting parameter $\alpha$ (**Lines 12-14**). For the last word, we rely solely on the generator's scores. To address length normalization issues when combining scores from both models, we re-normalize the merged score for the previous words by multiplying it by the length of the previous word tokens $j$ from the generator, adding the last word's score, and normalizing by the total length $n$ (**Lines 15-16**).

This integration process ensures that the re-rankers are utilized at the appropriate decoding stages, thereby enhancing the overall quality of the generated sequences by combining the strengths of both models.

## 3 Test Suites

The major advantage of combining models with different vocabularies zero-shot is that it leverages the strengths of available pre-trained models to generate more accurate and robust output. This is particularly relevant in multimodal scenarios, where unimodal systems excel in their respective modalities but are weaker or incapable of processing other modalities. Furthermore, it can also enhance quality compared to N-best list re-ranking when used as an ensembling technique as it waits until the complete sequence is generated. Hence, to validate our approach, we consider three MT scenarios as a test bed where quality can be improved by combining different sources and evaluating with targeted test sets that require information from both models. An overview of test suites is provided in Table 1.

## 3.1 Unimodal MT

We evaluate the use case of ensembling different LLM models to enhance translation quality. This is particularly relevant given the rapid development of various translation LLMs, where combining different systems can improve quality and robustness. We use the WMT 2022 *English → German* test set (Kocmi et al., 2022) to validate our approach and focus solely on assessing translation quality.

| Test Set | Language Pair | # Examples | Phenomena |
|---|---|---|---|
| MuST-SHE | *En → Fr* | 315 (1108) | Gender Disambiguation Translation |
| CoMMuTE | *En → De* | 300 | Word Disambiguation Translation |
| WMT22 | *En → De* | 2037 | Translation |

Table 1: Overview of test suites. For MuST-SHE, 315 examples are utterances where information is available in audio. However, we use the full test set with other types of bias when reporting translation quality.

## 3.2 Multimodal MT

Translating from English to gender-marked languages is challenging when the source text lacks clear gender cues. To evaluate bias in current NMT systems, Bentivogli et al. (2020) developed the *MuST-SHE* test suite, which includes examples with varying forms of gender bias. This suite features cases where gender information is conveyed through audio cues, such as the speaker's voice.

While End-to-End ST systems can handle such cases, they often fall short compared to advanced translation LLMs (Agarwal et al., 2023). Therefore, we use MuST-SHE for *English → French* to investigate if combining ST and translation LLMs can improve translation quality and address gender ambiguity.

Similarly, images can assist in disambiguating text and enhancing translation quality. However, translation LLMs typically do not process images, and vision LLMs alone are inadequate for translation tasks. We combine these models to leverage their strengths for better image-aware translations.

Existing vision translation test sets often lack ambiguity, making image inputs unnecessary (Vijayan et al., 2024). To address this, Futeral et al. (2023) introduced CoMMuTE, which features ambiguous source sentences with two images and their translations. We use CoMMuTE for *English*

*→ German* translation in a generative framework to evaluate if images can enhance translations without compromising overall quality.

## 4 Results

This section presents the experiments conducted using our ensembling approach across various test suites. Since each test suite has a distinct experimental setup, we will address them individually. First, we will specify the models and evaluation metrics applied in each scenario. Then, we will present the results and highlight our main findings.

### 4.1 Ensembling for Improving Translations

**Models:** We aim to combine two models that excel in translation but possess different strengths. For this purpose, we chose the *Madlad-10B*[2], an encoder-decoder architecture trained on extensive parallel data, and *ALMA-13B-R*[3], a decoder model trained using contrastive preference optimization and selecting high quality data (Xu et al., 2024b).

**Metrics:** As the models that we would like to ensemble are high quality, we report with several neural metrics to reliably validate the improvements. For reference-based we report with COMET (Rei et al., 2022a) and BLUERT (Sellam et al., 2020; Pu et al., 2021) whereas for reference-free we report with COMET-KIWI (Rei et al., 2022b), COMET-KIWI-XXL (Rei et al., 2023) and XCOMET-XXL (Guerreiro et al., 2023) metrics.

**Hyper-parameters:** We set the re-ranking weight $\alpha$ to $0.5$ given that both models have high quality and should be weighted equally. Furthermore, we set the $topk$ to 5 and the number of beams for the generator as 5.

To validate our combined model and online re-ranking approach, we compare it against several baselines. First, we check if the ensemble outperforms each individual model. Next, we evaluate if our method surpasses offline re-ranking techniques, indicating a more effective ranker influence and improved search space exploration during decoding.

We evaluate our approach using N-best list re-ranking, with *Madlad* as the generator and *Alma* as the ranker. We generate an N-best list of 25 hypotheses with $\alpha$ set to 0.5 to facilitate a fair

---

[2] https://huggingface.co/google/madlad400-10b-mt

[3] https://huggingface.co/haoranxu/ALMA-13B-R

| Generator | Ranker | Online | COMET 22 | COMET KIWI 22 QE | COMET KIWI XXL QE | XCOMET-XXL | BLEURT |
|---|---|---|---|---|---|---|---|
| | | | *No re-ranking* | | | | |
| GPT-4 | × | N/A | 87.29 | 83.48 | 84.91 | 97.56 | _ |
| Madlad-10B | × | N/A | 86.60 | 83.14 | 82.65 | 96.77 | 76.79 |
| Alma-13B-R | × | N/A | 86.40 | 83.28 | 84.25 | 97.48 | 77.20 |
| | | | *Offline re-ranking* | | | | |
| Madlad-10B | Alma-13B-R | × | 87.27 | 83.68 | 84.11 | 97.12 | 77.66 |
| Madlad-10B, Alma-13B-R | Madlad-10B, Alma-13B-R | × | 87.54 | **83.95** | 84.97 | 97.39 | 78.20 |
| | | | *Online re-ranking (ours)* | | | | |
| Madlad-10B | Alma-13B-R | ✓ | **87.69** | 83.94 | **85.20** | **97.68** | **78.36** |

Table 2: Performance of models on the WMT 22 *English → German* test set. Scores are highlighted in **bold** if it is the best in all configurations. Results for GPT-4 and Alma-13B-R are reported from (Xu et al., 2024b)
.

| Model | COMET 22 | COMET KIWI 22 QE | COMET KIWI XXL QE | XCOMET-XXL | BLEURT |
|---|---|---|---|---|---|
| GPT-4 | 87.29 | 83.48 | 84.91 | 97.56 | _ |
| Madlad | 86.60 | 83.14 | 82.65 | 96.77 | 76.79 |
| (Madlad) 5-best + QE | 87.33 | 83.83 | 86.45 | 97.25 | 77.78 |
| (Madlad + Alma Online re-rank) 5-best + QE | **87.66** | **84.12** | **87.86** | **97.91** | **78.31** |

Table 3: Performance of models on the WMT 22 *English → German* test set with Quality Estimation based re-ranking via selecting from 5-best list using comet-kiwi-xxl. Scores are highlighted in **bold** if it is the best in all configurations.

.

comparison between offline and online re-ranking methods. Additionally, we test a scenario where the N-best lists from both models are concatenated and jointly re-ranked on 50 hypotheses. We reports the results for the baselines and our approach in Table 2.

**Ensembling enables to reach state-of-the-art quality:** Both *Madlad* and *Alma* produce high-quality translations, though they still lag behind GPT-4 across all metrics. However, after applying offline re-ranking, their performance improves consistently, becoming competitive with GPT-4. When using our online re-ranking approach, the ensemble outperforms GPT-4 across all metrics and shows our proposed approach can improve the translation quality by a substantial margin.

**Online re-ranking outperforms offline joint re-ranking:** When *Madlad* serves as the generator and *Alma* as the ranker in our approach, the results are superior to those achieved with joint re-ranking, where both models are used simultaneously. Our approach enhances knowledge sharing and collaboration during the decoding process, leading to better translation quality.

### 4.1.1 Quality of N-best list

The primary motivation behind our approach was to influence the decoding process in real-time, rather than waiting until the end. If this is effective, we expect the N-best list to improve with online re-ranking. Additionally, using quality estimation should enhance the selection of the best hypothesis from the N-best list. To validate this, we utilize COMET-KIWI-XXL for selecting the best candidate from the top 5 beams of *Madlad*, comparing scenarios with and without online re-ranking and report the scores in Table 3.

We observe that integrating quality estimation significantly enhances Madlad's performance across all metrics. Using COMET-KIWI-XXL to select the best candidate from the top 5 beams improves score from $82.65 \rightarrow 86.45$. This improvement is also evident in the BLUERT score, increasing from $76.79 \rightarrow 77.78$. Additionally, comparing the top 5 beams with our approach, we find that the quality is superior, demonstrating that the early influence of *ALMA* in decoding. Furthermore, this allows to integration of multiple NMT models to generate the N-best list together and later combined with quality estimation for maximum performance.

## 4.2 Speech-Aware Translations

**Models:** To tackle gender ambiguity in text translation using speaker voice information, we combine a robust text translation model with a speech-based model that excels at disambiguating gender, even if it is not as strong in translation. We use the *Madlad* model (Kudugunta et al., 2024) for high-quality text translation with gold transcript and the *Seamless*[4] model for speech translation. Our approach employs *Madlad* as the *generator* and *Seamless* as the *ranker*, allowing us to leverage the speech model's ability to correct gendered forms in the translation.

However, we observed that the *Seamless* model exhibited a bias toward the masculine gender and struggled to effectively resolve gender ambiguities using speech. To mitigate this, we conducted additional fine-tuning using LoRA (Hu et al., 2021) on a balanced speaker dataset derived from MuST-C (TED talks) with gender annotations (Di Gangi et al., 2019; Gaido et al., 2020) (Training details in Appendix A.1). We remove talks that are present in MuST-SHE for no overlap. This "debiasing" process improved the model's ability to disambiguate gender based on speech. Consequently, we use the *Madlad* and adapted *Seamless* models to generate high-quality, speech-aware translations.

**Metrics:** To evaluate the effectiveness of our approach in disambiguating gender and improving translation quality, we use several key metrics. For gender disambiguation, we follow the methodology of Bentivogli et al. (2020) and report two metrics: accuracy (correct gender form is present) and coverage (either gender form is present).

For overall translation quality, we report BLEU (Papineni et al., 2002), ChrF2 (Popović, 2016) calculated using SacreBLEU (Post, 2018), and COMET (Rei et al., 2022a) (*wmt22-comet-da*) for brevity.

Additionally, we report *Sensitivity*, which measures the difference between the scores of correctly and incorrectly gendered references, as suggested by Bentivogli et al. (2020).

**Hyper-parameters:** For decoding with *Madlad*, we use beam search with 5 beams. Our proposed algorithm involves two key parameters: $\alpha$ and $topk$.

---

We set $topk$ to 5, resulting in a total of 25 candidates being ranked by *Seamless* at each step.

We optimized $\alpha$ through grid search on the MuST-C development set (Appendix A.3) via offline re-ranking and setting it to 0.8 based on these results. We also create an N-best list of 25 hypotheses with $\alpha$ at 0.8 for offline comparison and perform joint re-ranking on the combined 50 N-best lists. Results are summarized in in Table 4.

***Madlad* and *Seamless* complement each other:** *Madlad* excels in overall translation quality (83.5) compared to *Seamless* (79.31). While *Seamless* initially favors masculine terms, fine-tuning on balanced data improves overall quality to 80.48, significantly reducing masculine bias (90.44 to 65.89) and increasing feminine representation (25.92 to 50.18). Thus, the adapted *Seamless* demonstrates improved gender disambiguation, though *Madlad* remains superior in overall translation. Hence, combining the models can be highly beneficial.

**Online re-ranking improves overall translation quality:** After re-ranking with N-best list, we see that the translation quality is improved when *Madlad* as a generator and *Seamless Bal* as a ranker model (83.50 → 83.66). In the opposite scenario where *Seamless Bal* uses *Madlad* as a ranker model, the quality also improves (80.48 → 81.31) but is lower than *Madlad* alone. However, during online re-ranking, we see that we achieved the best performance of 83.78. This suggests that our approach facilitates knowledge sharing between the models during decoding, leading to significant quality enhancements.

**Balance between translation quality and gender disambiguation through online re-ranking** We observe that the highest accuracies for feminine terms (1F) are achieved when *Seamless Bal* is employed as a generator. Nevertheless, the overall translation quality in these instances is considerably lower compared to scenarios where *Madlad* is the generator. By using *Madlad* as a generator, we attain a higher average 1F score of 60.32 compared to offline re-ranking without compromising overall translation quality and better distribution across gender. Moreover, we achieved the highest sensitivity score of 1.1 across all configurations. This shows that our approach can consistently perform better than traditional N-best list re-ranking.

While the scores for the disambiguation are not

| Generator | Ranker | Online | 1F (Acc %) | 1F (Term Cov %) | 1M (Acc %) | 1M (Term Cov %) | Avg (Acc %) | COMET Correct | △ |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *No re-ranking* | | | | | |
| Madlad | × | N/A | 25.92 | 68.39* | 90.44* | 63.65* | 58.18 | 83.52 | 0.90 |
| Seamless | × | N/A | 20.28 | 63.20 | 88.30 | 62.43 | 54.29 | 79.31 | 0.73 |
| Seamless Bal | × | N/A | 50.18 * | 62.73 | 65.89 | 59.02 | 58.03 | 80.48 | 0.83 |
| | | | | *Offline re-ranking* | | | | | |
| Madlad | Seamless Bal | × | 28.81 | 67.92 | **89.59** | 63.41 | 59.20 | 83.66 | 0.96 |
| Seamless Bal | Madlad | × | **40.90** | 65.09 | 77.99 | 60.97 | 59.44 | 81.31 | 0.90 |
| Madlad, Seamless Bal | Madlad, Seamless Bal | × | 29.83 | 67.92 | **89.59** | 63.41 | 59.71 | 83.64 | 0.96 |
| | | | | *Online re-ranking (ours)* | | | | | |
| Madlad | Seamless Bal | ✓ | 33.78 | **68.16** | 86.86 | **63.65*** | **60.32*** | **83.78*** | **1.1*** |

Table 4: Performance of models on the MuST-SHE test set for speech-aware translations. *Seamless Bal* indicates the adapted model trained on balanced gender data. △ denotes the *sensitivity*, i.e., the difference in scores between correct and incorrect references. Scores are highlighted in **bold** if online re-ranking improves over offline re-ranking and ∗ if it is the best in all configurations.

| Generator | Ranker | Online | BLEU Correct | △ | Chrf2 Correct | △ | COMET Correct | △ |
|---|---|---|---|---|---|---|---|---|
| Madlad-10B | × | N/A | 45.9 | 0.4 | 62.3 | 1.3 | 82.90 | 0.06 |
| PaliGemma-3B MT | × | N/A | 27.6 | **5.7** | 51.0 | **7.3** | 79.58 | **8.25** |
| Madlad-10B | PaliGemma-3B MT | × | 46.1 | 1.9 | **62.6** | 1.7 | **83.45** | 1.17 |
| Madlad-10B | PaliGemma-3B MT | ✓ | **46.2** | 1.8 | **62.6** | 1.9 | 83.25 | **1.34** |

Table 5: Performance of models on the CoMMuTE *English → German* test set for image-aware translations. △ indicates the sensitivity i.e difference between correct and incorrect references. Scores are highlighted in **bold** if it is the best in all configurations.

.

high, we would like to highlight that we focused on combining the strengths of the models. However, one can use targeted systems such as Gaido et al. (2020) to further improve the performance for the desired tasks.

## 4.3 Image-Aware Translations

**Models:** To integrate image information for disambiguating source text, a robust multimodal machine translation (MT) system is essential. Initially, we experimented with the off-the-shelf instruction-tuned Llava model[5] (Li et al., 2024). While Llava provided reasonable results, its performance was sub-par for our needs. Consequently, we chose to fine-tune the *PaliGemma* model[6] (Beyer et al., 2024), which was originally trained to generate captions in multiple languages. We fine-tuned *PaliGemma* using the Multi30k image captions

[5] https://huggingface.co/llava-hf/llava-v1.6-vicuna-13b-hf
[6] https://huggingface.co/google/paligemma-3b-ft-cococap-448

dataset (Elliott et al., 2016), adapting it with Q-LoRA (Appendix A.2) for enhanced image-aware translations (*PaliGemma-3B MT*).

**Metrics:** For evaluating this task, we use BLEU, ChrF2, and COMET scores, as we do not have specific annotations for words in the target sentences. To assess the impact of contextual information provided by the images, we also report the sensitivity metric △, to estimate how much the image context influences the translation quality.

**Hyper-parameters:** Vision LLMs require more memory because the image is encoded into a long sequence of tokens. Consequently, we were limited to using a beam size of 3 with a top-k of 3. Additionally, tuning the parameter $\alpha$ was challenging due to the lack of a dedicated ambiguous test set; using a standard test set would result in no weight being given to the vision model. Therefore, we report the oracle $\alpha$ of 0.9, which represents the best-performing weight on the test set, determined through a grid search with offline re-ranking. We report the scores in Table 5.

**PaliGemma is highly sensitive to image context:** We observe that the sensitivity $\triangle$ of our fine-tuned *PaliGemma* model for MT is notably high across all metrics (e.g., 5.7 BLEU), demonstrating that the model is effectively using the image information to influence its translations. This suggests that *PaliGemma* does not disregard the visual context during translation. However, despite this sensitivity, *PaliGemma's* overall translation quality significantly lags behind that of *Madlad*, as indicated by the lower COMET score (difference of 3.32). This disparity highlights the potential benefit of combining the strengths of both models to achieve more accurate and image-aware translations.

**No clear winner between offline and online re-ranking:** Comparing offline and online re-ranking, we find that re-ranking with *PaliGemma* enhances translations, evidenced by a sensitivity $\triangle$ increase of up to 1.28 COMET. There's also a slight improvement in overall translation quality after re-ranking. However, the difference between the two approaches is modest, especially given the small test set size of 300 examples.

We hypothesize two main factors behind the results. First, *Madlad* assigns very low probabilities to translations of ambiguous words it isn't biased toward, while *PaliGemma* avoids extremely high probabilities. As a result, merging probabilities tends to favor the incorrect translation with the highest overall score. Second, the test sentences are short, averaging 4-5 words, so the N-best list includes diverse variations, making offline re-ranking similar to the online approach. However, we believe our online re-ranking method could benefit longer sentences and stronger vision translation models.

## 5 Related Work

**Fusion for MT:** Integrating additional language models into MT systems via shallow or deep fusion, or through re-ranking, to improve translation quality is a well-studied area (Chen et al., 2006; Hasan et al., 2007; Gulcehre et al., 2015; Li and Jurafsky, 2016; Gulcehre et al., 2017; Herold et al., 2023). Stahlberg et al. (2018) explored advanced fusion method where an NMT model is trained from scratch while keeping a pre-trained language model fixed, allowing the model to learn only what

is missing. There has also been growing interest in combining NMT with document-level language models (Stahlberg et al., 2019; Petrick et al., 2023; Hoang et al., 2024). Unlike previous works that utilize static weights for merging probabilities, Jean and Cho (2020) propose dynamic coefficients, which are crucial for effectively combining models with different strengths.

**Ensembling:** System combination, which involves merging multiple hypotheses to generate a better version, is one approach to leveraging the strengths of different models (Bangalore et al., 2001; Matusov et al., 2006; Heafield and Lavie, 2010; Freitag et al., 2014). Another approach is to merge model parameters (Junczys-Dowmunt et al., 2016) or distill knowledge from the models (Freitag et al., 2017). With the increasing diversity of LLMs, recent research has explored methods to combine them through vocabulary merging (Xu et al., 2024b), generating new outputs based on hypotheses (Jiang et al., 2023b), or dynamically selecting different models at each step (Shen et al., 2024).

Our work differs from these approaches as it neither relies on vocabulary matching nor requires additional training data.

## 6 Conclusion

We proposed a novel ensembling strategy that operates at the word level during the decoding process to enhance knowledge sharing. Our approach demonstrated significant benefits across multiple scenarios. It proved effective for ensembling translation systems, and even when combined with quality estimation models, it achieved state-of-the-art translation quality. Additionally, experiments on targeted multimodal test sets revealed that our method facilitates better knowledge sharing compared to traditional re-ranking techniques.

For future work, we propose to explore unsupervised dynamic selection, enabling models to generate outputs only when they are better equipped for the task. We believe this approach could address the current limitations and lead to more significant improvements in image-aware translation.

## 7 Limitations

The major limitation of this work is that we operate at word-level which is not compatible for several

languages that are character based. Hence, it is not trivial to merge models for generating such languages. Further analysis is necessary on character-level tokenization to accurately re-rank during the decoding steps.

Another drawback is that, although re-ranking enhances translation quality, it incurs a latency cost. Unlike offline re-ranking, our approach employs the ranker model at each time step, resulting in significantly slower performance.

Finally, we focused mainly on ensembling the two models using static weights. However, since the models have different strengths, it is crucial to determine when to rely on one model or ensemble both. This dynamic approach would better exploit each model's strengths while avoiding the integration of their weaknesses.

## References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, et al. 2023. Findings of the iwslt 2023 evaluation campaign. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Bangalore Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.*, pages 351–354. IEEE.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, et al. 2024. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*.

Boxing Chen, Roldano Cattoni, Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. 2006. The itc-irst smt system for iwslt 2006. In *Proceedings of the Third International Workshop on Spoken Language Translation: Evaluation Campaign*.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open source machine translation system combination. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32.

Matthieu Futeral, Cordelia Schmid, Ivan Laptev, Benoît Sagot, and Rachel Bawden. 2023. Tackling ambiguity with images: Improved multimodal machine translation and contrastive evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5394–5413, Toronto, Canada. Association for Computational Linguistics.

Marco Gaido, Beatrice Savoldi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2020. Breeding gender-aware direct speech translation systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3951–3964.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *arXiv preprint arXiv:2310.10482*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148.

Saša Hasan, Richard Zens, and Hermann Ney. 2007. Are very large n-best lists useful for smt? In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 57–60.

Kenneth Heafield and Alon Lavie. 2010. Combining machine translation output with open source. *The carnegie mellon multi-engine machine translation scheme. The Prague Bulletin of Mathematical Linguistics*, 93(1):27–36.

Christian Herold, Yingbo Gao, Mohammad Zeineldeen, and Hermann Ney. 2023. Improving language model integration for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7114–7123.

Hieu Hoang, Huda Khayrallah, and Marcin Junczys-Dowmunt. 2024. On-the-fly fusion of large language models and machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 520–532.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Sébastien Jean and Kyunghyun Cho. 2020. Log-linear reformulation of the noisy channel model for document-level neural machine translation. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 95–101.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 319–325.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 2024. Preliminary wmt24 ranking of general mt systems and llms. *arXiv preprint arXiv:2407.19884*.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, et al. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2024. Llavamed: Training a large language-and-vision assistant

for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–40.

GemmaTeam Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Benjamin Minixhofer, Edoardo Maria Ponti, and Ivan Vulić. 2024. Zero-shot tokenizer transfer. *arXiv preprint arXiv:2405.07883*.

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.

Frithjof Petrick, Christian Herold, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2023. Document-level language models for machine translation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 375–391, Singapore. Association for Computational Linguistics.

Maja Popović. 2016. chrf deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Amy Pu, Hyung Won Chung, Ankur P Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for mt. In *Proceedings of EMNLP*.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022a. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Ricardo Rei, Nuno M Guerreiro, Daan van Stigt, Marcos Treviso, Luísa Coheur, José GC de Souza, André FT Martins, et al. 2023. Scaling up cometkiwi: Unbabel-ist 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848.

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC de Souza, Taisiya Glushkova, Duarte Alves, Luísa Coheur, et al. 2022b. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 204–211.

Felix Stahlberg, Danielle Saunders, Adrià de Gispert, and Bill Byrne. 2019. Cued@ wmt19: Ewc&lms. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 364–373.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, MA Zejun, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vipin Vijayan, Braeden Bowen, Scott Grigsby, Timothy Anderson, and Jeremy Gwinnup. 2024. The case for evaluating multimodal translation models on text datasets. *arXiv preprint arXiv:2403.03014*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Yangyifan Xu, Jinliang Lu, and Jiajun Zhang. 2024b. Bridging the gap between different vocabularies for llm ensemble. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7133–7145.

# A Appendix

## A.1 Adapting Seamless

We use the gender annotations from Gaido et al. (2020) to select talks with feminine speaker pronouns and an equal amount of randomly sampled masculine talks that are in the training set. We use the huggingface transformer's library (Wolf et al., 2019) for fine-tuning Seamless. We use LoRA (Hu et al., 2021) to fine-tune Seamless on this data. We set the *rank* to 16, *lora_alpha* to 64 and *lora_dropout* to 0.1. We apply adapters on the following modules: *q_proj, v_proj, linear_q, linear_v*. We set *batch_size* to 16, *gradient_accumulation_steps* to 8 and train with *fp16* for 20 epochs validating at every 200 steps. The *learning_rate* is set to $1e^{-5}$. The other parameters are set to default in the transformers library.

## A.2 Adapting PaliGemma

We also fine-tune the PaliGemma model with the huggingface transformer's library (Wolf et al., 2019) but use Q-LoRA (Dettmers et al., 2023) with *4-bit* quantization as the vision models require more VRAM. We set the *rank* to 8, *lora_alpha* and *lora_dropout* to default. We apply adapters on the following modules: *q_proj, k_proj, v_proj, gate_proj, up_proj, down_proj*. We set *batch_size* to 2, *gradient_accumulation_steps* to 6 and train with *bf16* for 5 epochs validating at every 200 steps. The *learning_rate* is set to $2e^{-5}$ with *AdamW* optimizer. The other parameters are set to default in the transformers library.

## A.3 Hyper-parameter Tuning for Speech-Aware Translations

To find the re-ranking weight $\alpha$, we generate the 25-best list of *Madlad* and *Seamless* on the MuST-C development set. Then, we calculate the scores of the models on these hypothesis and perform a grid search to find the optimal weight. Here, $\alpha = 1$ means that the score is only from *Madlad* and $\alpha = 0.5$ means equal contribution. The grid search is plotted in Figure 2.

We see that $\alpha$ as 0.8 is always achieving higher scores. Furthermore, we see that using *Seamless* as generator (Figure 2b) leads to poor translation quality and $\alpha$ as 1. However, in the case of *Madlad* as a generator (Figure 2a), we see that $\alpha$ as 1 is not optimal showing that re-ranking with *Seamless* is indeed beneficial. Finally in the case of both models as generator (Figure 2c), we again see that $\alpha$ as 1 achieves highest quality showing that *Seamless* is not beneficial.
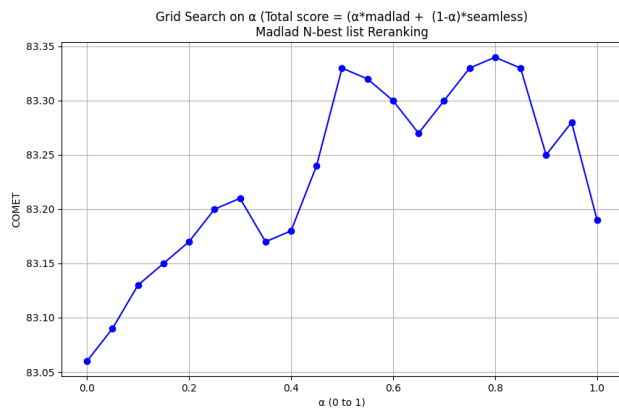
(a) Re-ranking with Madlad N-best list
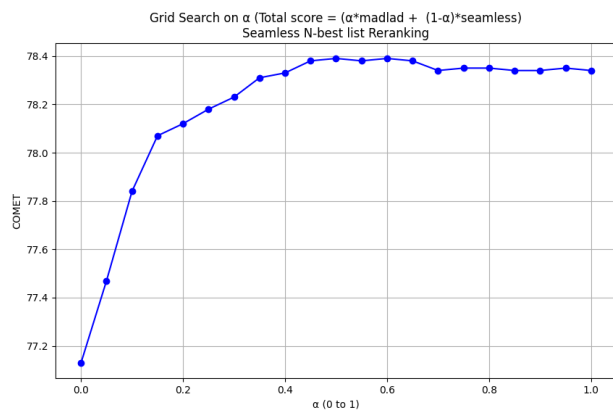


(b) Re-ranking with Seamless N-best list
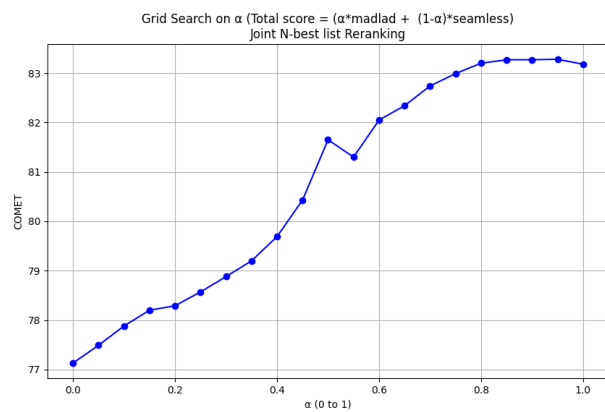


(c) Re-ranking with Joint N-best list

Figure 2: Grid Search on $\alpha$ with *Madlad* and *Seamless Bal* on the MuST-C development set with N-best lists from different generators and rankers.