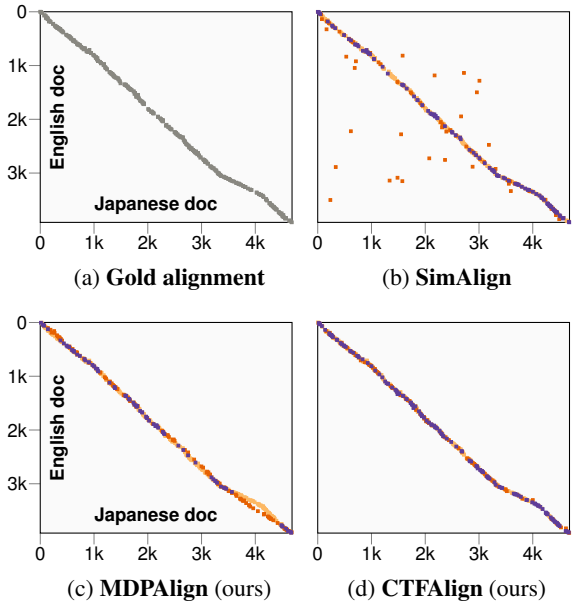
Scaling Unsupervised Word Alignment to Documents via Structural Constraints

Michelle Wastl Jannis Vamvas Rico Sennrich
Department of Computational Linguistics, University of Zurich
{michelle.wastl,jannisnikos.vamvas,rico.sennrich}@uzh.ch

Abstract

Word alignment has traditionally been studied between sentences, but many cross-lingual tasks increasingly require correspondences across full documents. While recent multilingual embedding models can encode long inputs, we show that applying algorithms designed for sentences directly to documents leads to performance degradation. To address this, we introduce **CTFAlign**, a lightweight, training-free approach for document-level word alignment. CTFAlign applies a **c**oarse-**t**o-**f**ine refinement strategy that restricts the alignment search space to semantically similar regions. Additionally, we introduce **MDPAlign**, a simpler alternative that constrains alignments by position with a **m**ain **d**iagonal **p**rior. Both approaches operate directly on full documents without relying on sentence segmentation or sentence alignment. We evaluate these methods across six language pairs varying in typological distance, resourcedness, and document length. Averaged over three models, CTFAlign reduces word alignment error rate from 0.412 to 0.326. These gains transfer downstream, leading to improvements in document-level translation coverage evaluation and recognition of semantic differences. We release CTFAlign as a Python package and make the code and data to reproduce our experiments publicly available.[1]



(a) **Gold alignment** (b) **SimAlign** (c) **MDPAlign** (ours) (d) **CTFAlign** (ours)

Figure 1: Performing word alignment on a pair of English and Japanese Wikipedia articles shows the differences between alignment algorithms and the correct alignments, false negatives, and false positives they predict. SimAlign (Jalili Sabet et al., 2020) produces some stray false positives that are implausible given the structure of the documents. MDPAlign focuses on the diagonal but misses the bend towards the end; CTFAlign is closest to the gold alignment. The figure is based on `Qwen3-Embedding-4B` embeddings and subsamples 10% of points for visualization. The two axes are the token positions in the document pair.

1 Introduction

Word alignment is a foundational task in natural language processing with the aim to identify semantically corresponding words between parallel texts. The task has applications such as hallucination detection (Zhou et al., 2021; Li et al., 2022; Huang et al., 2024), label projection (Ehrmann et al., 2011; García-Ferrero et al., 2022; Chen et al., 2023; Fetahu et al., 2023), machine translation (Kong et al., 2019; Wu et al., 2024; Koshkin et al., 2024; Wang et al., 2025), and cross-lingual representation learning (Miao et al., 2024).

Most of these tasks have moved beyond the sentence level with large language models (LLMs) and embedding models being capable of processing entire documents in a single pass (Zhang et al., 2024; Marone et al., 2025; Boizard et al., 2025; Zhang et al., 2025). In contrast, word alignment methods

[1]Package: https://github.com/ZurichNLP/CTFAlign; experimental code: https://github.com/ZurichNLP/document-level-word-alignment; data: https://huggingface.co/datasets/ZurichNLP/document-level-word-alignment

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 7419–7436
October 24-29, 2026 ©2026 Association for Computational Linguistics

still largely operate on sentence pairs (Dou and Neubig, 2021; Wu et al., 2023; Huang et al., 2024; Miao et al., 2025), despite the fact that embedding-based approaches such as SimAlign (Jalili Sabet et al., 2020) are technically applicable to full documents.

However, when sentence-level word alignment algorithms are directly applied to documents, performance degrades as similarity matrices become larger and noisier (Figure 1b). In this work, we investigate how embedding-based word alignment can be extended from sentences to full documents without relying on sentence segmentation or additional supervision.

Our contributions are summarized as follows:

- We introduce two lightweight, model-agnostic methods for document-level word alignment: MDPAlign (Figure 1c) and CTFAlign (Figure 1d).
- Our approaches reduce the degradation caused by scaling sentence-level word alignment to full documents, with CTFAlign in particular matching sentence-level performance in several settings.
- We show that our document-level word alignment approaches transfer to downstream tasks, improving translation coverage evaluation and token-level recognition of semantic differences.

2 Background & Related Work

2.1 Word Alignment

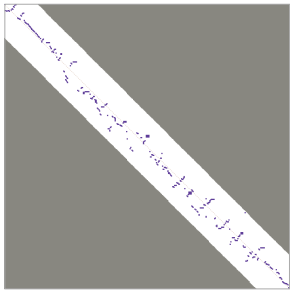
Modern word alignment approaches typically leverage neural networks to solve this task with varying levels of supervision: SimAlign (Jalili Sabet et al., 2020) showed that contextual embeddings from multilingual encoders can be leveraged to produce competitive word alignments without further training by applying similarity-based matching algorithms. Dou and Neubig (2021) subsequently showed that embedding-based word alignment algorithms perform better if the underlying encoder is fine-tuned with parallel data. Arase et al. (2023) instead formulate embedding-based word alignment as an optimal transport problem, including partial and unbalanced variants that account for null alignments. WSPAlign (Wu et al., 2023) proposed to include supervision signals from parallel, word-aligned as well as noisy data, and BinaryAlign (Latouche et al., 2024) framed word alignment as a binary sequence labeling task. Most recently, word alignment has been approached by fine-tuning LLMs on word alignment labeled data (Miao et al., 2025). These methods consistently focus on word alignment between sentences.

Word alignment between documents remains comparatively underexplored. In practice, for longer texts the task would have to be broken down into multiple steps by segmenting texts either into sentences or chunks and aligning them before applying sentence-level word alignment. The dependence on sentence segmentation and sentence alignment can be brittle, especially for languages or domains where reliable segmentation and alignment tools are unavailable. It may also discard document-level information and introduce cascading errors before word alignment is even performed. Even where such a pipeline remains usable despite imperfect intermediate steps, it introduces additional engineering and maintenance overhead: embedding-based pipelines require segmentation, sentence representation and alignment, followed by token representation and word alignment. In contrast, with long-context encoders, our approach obtains token representations for the full document in a single encoder pass and performs alignment directly on the similarity matrix.

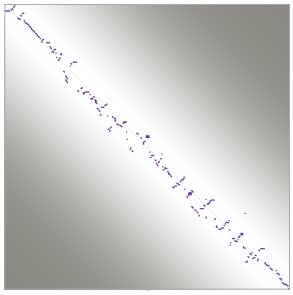
2.2 Positional Constraints and Priors in Alignment

Alignment methods have long relied on positional structure and ordering assumptions. Based on the observation that translations tend to preserve the order of the content, alignment was conditioned on relative token positions for IBM Model 2 (Brown et al., 1993; Dagan et al., 1993). FastAlign (Dyer et al., 2013) later simplified this idea into an efficient diagonal positional prior tailored to sentence-level word alignment, and SimAlign (Jalili Sabet et al., 2020) proposed a distortion correction that penalizes large deviations in relative position. Related assumptions have also been used in sentence alignment. Gale and Church (1993) exploited the near-monotonic structure of parallel documents by constraining alignment paths to remain diagonal.

MDPAlign extends this intuition from sentence and sentence-alignment settings to document-level word alignment by imposing a global diagonal prior.



(a) Strict MDPAlign with bandwidth $k = 50$.



(b) Fuzzy MDPAlign masking with gradual decay away from the diagonal with $k = 150$.

Figure 2: Illustration of the two MDPAlign variants. Both assume similar relative positions of corresponding words across documents and constrain alignment search toward the main diagonal. Alignments shown in search space with darkness of gray indicating suppression of off-diagonal similarities.

2.3 Coarse-to-Fine Methods

Coarse-to-fine strategies have been widely used across NLP to reduce large search spaces by progressively refining predictions (Petrov et al., 2008; Petrov, 2011; Dong and Lapata, 2018). Closest to our work is VecAlign (Thompson and Koehn, 2019), a sentence alignment method that recursively narrows the alignment search space within a document using sentence embeddings. However, VecAlign operates on sentence representations and searches for a monotonic, contiguous alignment path with dynamic programming. In contrast, CTFAlign applies coarse-to-fine refinement directly to token-level similarity matrices and remains compatible with non-monotonic, non-contiguous alignment patterns at all resolutions.

3 Methods

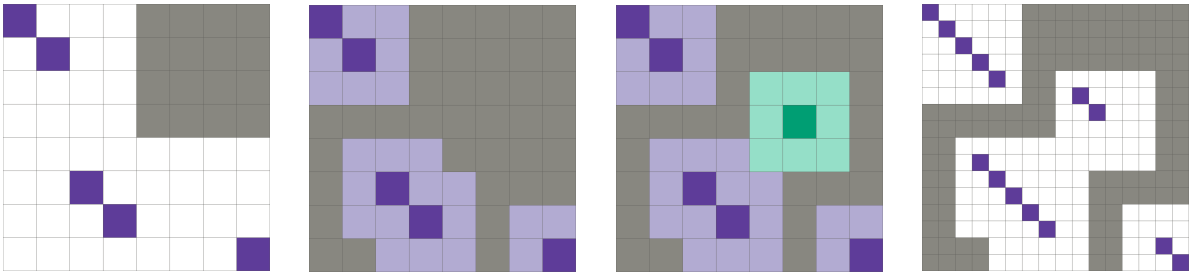
3.1 Document-level Word Alignment

Task description Given a parallel document pair consisting of a source word sequence $\mathbf{x} = (x_1, \ldots, x_m)$ and a target word sequence $\mathbf{y} = (y_1, \ldots, y_n)$, the goal of document-level word alignment is to produce a set of alignment links $\mathcal{A} \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ connecting semantically corresponding words across the two documents. Unlike sentence-level word alignment, documents may contain thousands of words, sentence boundaries may be unavailable, and lexical repetitions can induce many plausible but incorrect matches across the whole document. As a result, the alignment search space for documents becomes noisier and less locally constrained than for sentences.

Embedding-based alignment We build on SimAlign (Jalili Sabet et al., 2020), which extracts contextual token representations from a multilingual encoder and computes a cosine-similarity matrix $S \in \mathbb{R}^{m \times n}$, where each entry S_{ij} represents the similarity between source token x_i and target token y_j. Alignment links are then predicted based on S using an alignment algorithm such as Argmax or Itermax (Jalili Sabet et al., 2020). Possible but rare negative cosine similarities are not considered for alignments. Both identify high-similarity token pairs bidirectionally: Argmax does so in a single pass, leading to sparser alignments, while Itermax applies the procedure iteratively to recover additional links. The token-level alignments are then converted to word alignments by mapping each subword token back to its parent word and treating two words as aligned whenever at least one of their token pairs is linked. While effective at the sentence level, if applied without constraint, these algorithms become susceptible to noise in document-length similarity matrices and lead to spurious alignments, as shown in Figure 1b.

Structural constraints for document-level alignment Our central hypothesis is that the primary challenge in document-level word alignment lies in the unconstrained search space of the similarity matrix. We therefore propose to constrain the alignment search space by applying structural constraints directly to the similarity matrix before applying an alignment algorithm.

We introduce two mechanisms for constraining the document-level word alignment search space. First, **MDPAlign** (Subsection 3.2; Figure 2) builds on the assumption that translations often preserve global structure, such that corresponding content occupies approximately the same relative position in both documents. It encodes this prior through diagonal-band masking, biasing alignments toward



(a) Detection of high-similarity blocks in a coarse-grained grid using average-pooled cosine similarities. The upper-right region was masked at the previous resolution.

(b) Expansion of detected blocks with a buffer of $w = 1$ to neighboring regions.

(c) Recovery of blocks with fully empty rows and columns using a buffer of $w = 1$.

(d) Refinement at higher resolution by halving block size and repeating the search in the remaining unmasked regions.

Figure 3: Illustration of one iteration of the CTFAlign algorithm. White regions denote searchable grid space, while gray regions are masked because they fall outside the aligned coarse blocks identified at the previous resolution. The process repeats from an initial 2×2 grid down to the token-level $m \times n$ grid.

the main diagonal of the similarity matrix.

While this positional correspondence oftentimes holds for direct translations, realistically, documents that convey the same content may also exhibit reordering, insertions, and deletions. We therefore introduce as a second approach **CTFAlign** (Subsection 3.3; Figure 3), which performs adaptive coarse-to-fine refinement, recursively identifying likely alignment regions and restricting subsequent alignment to those regions, allowing the alignment structure to adapt throughout the document.

Both methods operate as lightweight, model-agnostic processing steps of the similarity matrix and require no further training of the underlying encoder.

3.2 Main Diagonal Prior (MDPAlign)

Let $S \in \mathbb{R}^{m \times n}$ be the token-level similarity matrix for a source of m tokens and a target of n tokens. As shown in Figure 2, we map source token i to its relative position i/m and target token j to j/n, and define their relative distance from the main diagonal as

$$d_{ij} = \left| \frac{i}{m} - \frac{j}{n} \right|. \tag{1}$$

We then apply one of two masking schemes, controlled by a bandwidth hyperparameter k. We anchor k to the longer document as an absolute token count rather than a fraction of document length, on the assumption that reordering in well-translated text is largely local (clause- or sentence-level) and does not grow with document length.

Strict We zero out all entries outside a diagonal band:

$$\tilde{S}_{ij} = \begin{cases} S_{ij}, & \text{if } d_{ij} \leq \frac{k}{\max(m,n)}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Fuzzy Since zeroing out all entries beyond a certain band is oftentimes too harsh, we alternatively apply a Gaussian decay, where k determines the standard deviation, preserving every cell but suppressing off-diagonal scores exponentially:

$$\tilde{S}_{ij} = S_{ij} \cdot \exp\left(-\frac{d_{ij}^2}{2\sigma^2} \right), \tag{3}$$

$$\sigma = \frac{k}{\max(m, n)}. \tag{4}$$

To the modified matrix $\tilde{S}$ we then apply the SimAlign (Jalili Sabet et al., 2020) alignment algorithm.[2]

3.3 Coarse-to-Fine Alignment (CTFAlign)

As visualized in Figure 3, we divide the similarity matrix S into a 2×2 coarse grid by average-pooling cosine similarities, where negative similarities count as zero, and run SimAlign (Jalili Sabet et al., 2020) on this coarse matrix. Each aligned block pair identifies a region of S likely to contain true alignments. We zero out all regions of S not

[2] (Jalili Sabet et al., 2020) also introduce a variant of SimAlign with a parabolic discount for distortion control. However, they report mixed results on the sentence-level, and we deem their parabolic discount, flatter than our Gaussian one, unsuitable for long documents.

covered by any aligned block. Formally, at each intermediate level t:

$$S_{ij}^{(t+1)} = \begin{cases} S_{ij}^{(t)}, & \text{if } (i,j) \in \mathcal{M}^{(t)}, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where $\mathcal{M}^{(t)}$ is the set of token pairs whose containing coarse block was retained at level t. We then halve the block size and repeat on $S^{(t+1)}$, roughly doubling the resolution at each step, until we reach token resolution, at which point the alignment produced on the final $m \times n$ grid is returned directly. If document lengths are skewed, token resolution is reached for one side earlier than for the other. In these cases, the shorter side is held at token resolution, while the other continues to be refined, so the process terminates only once both sides have reached token resolution.

Two mechanisms improve robustness against coarse-level errors. First, a width hyperparameter $w \geq 0$ controls how many neighboring blocks are retained around each aligned block at every resolution. Second, we apply a recovery step at each level: any block at the intersection of a source row and a target column, of which neither has any other retained block, is recovered. A recovered block also retains a buffer of width w additional blocks around it. A reference implementation of the CTFAlign algorithm is provided in Appendix A.

4 Experimental Setup

Data Since there are currently no datasets that include word alignment annotation directly for documents, we reconstruct documents from human-annotated sentence-level word alignment datasets that include information about document membership and sentence order. Hence, our evaluation includes English–French (en–fr; Mihalcea and Pedersen, 2003), English–Romanian (en–ro; Mihalcea and Pedersen, 2003), English–Japanese (en–ja; Neubig, 2011), English–Chinese (en–zh; Liu and Sun, 2015), English–Czech (en–cz; Mareček, 2011), and Latin–Greek (la–gr; Yousef et al., 2022)—covering six language pairs with varying degrees of typological distance and resourcedness. We use existing development/test splits where available. For la–gr, we split 25% into a development set and leave the remaining 75% as a test set, while en–cz is only considered as a test set. Statistics on datasets and splits are presented in Appendix B.1.

Models We generate contextualized token embeddings for entire document pairs using two recent multilingual long-context models that encode documents in a single forward pass: `Qwen3-Embedding-4B` (40,960 tokens; Zhang et al., 2025), a decoder-based embedding model contrastively trained on large-scale text pairs, including bitext-mining data, and `mmBERT-base` (307M parameters, 8,192 tokens; Marone et al., 2025), a general-purpose multilingual encoder pretrained with masked language modeling without explicit parallel-data or pairwise embedding supervision. Additionally, we include `LaBSE` (500M parameters; Feng et al., 2022), which is explicitly trained for cross-lingual similarity but has a more limited context window (512 tokens). In this case, we embed sentences individually and concatenate their representations. Model choice, layer selection, and hyperparameters are determined on the development set. Development results are reported in Appendix E. For language pairs without a development split (i.e., en–cz), we choose the layer that performed best most frequently across the remaining language pairs at the same granularity. We then compute token-level cosine similarity matrices from the resulting token embeddings.

Baselines We use SimAlign (Jalili Sabet et al., 2020) with Argmax as our baseline at both sentence- and document-level.[3]

The unconstrained document-level setting serves as our primary baseline for measuring the effect of document-scale alignment, while the sentence-level setting provides an approximate upper bound.

As an additional baseline, we use an LLM to autoregressively generate the word alignment labels given a simple 1-shot prompt without fine-tuning. Our prompt is based on the "full-mode" prompt defined by Miao et al. (2025), which generates alignment labels in one pass. We disregard their "marker mode" prompt, which would rely on as many passes as both documents contain words, deeming it unfeasible for long documents. We use `gpt-5.4-mini-2026-03-17` to generate the JSON output in OpenAI's structured output mode with a temperature of 0, while all other parameters are kept at default (see Appendix D for the prompt).

Implementation Details We evaluate both variants of MDPAlign (*strict* and *fuzzy*) as well as

[3]For our experiments with the development set, we also considered Itermax as a baseline. See Appendix E for details.

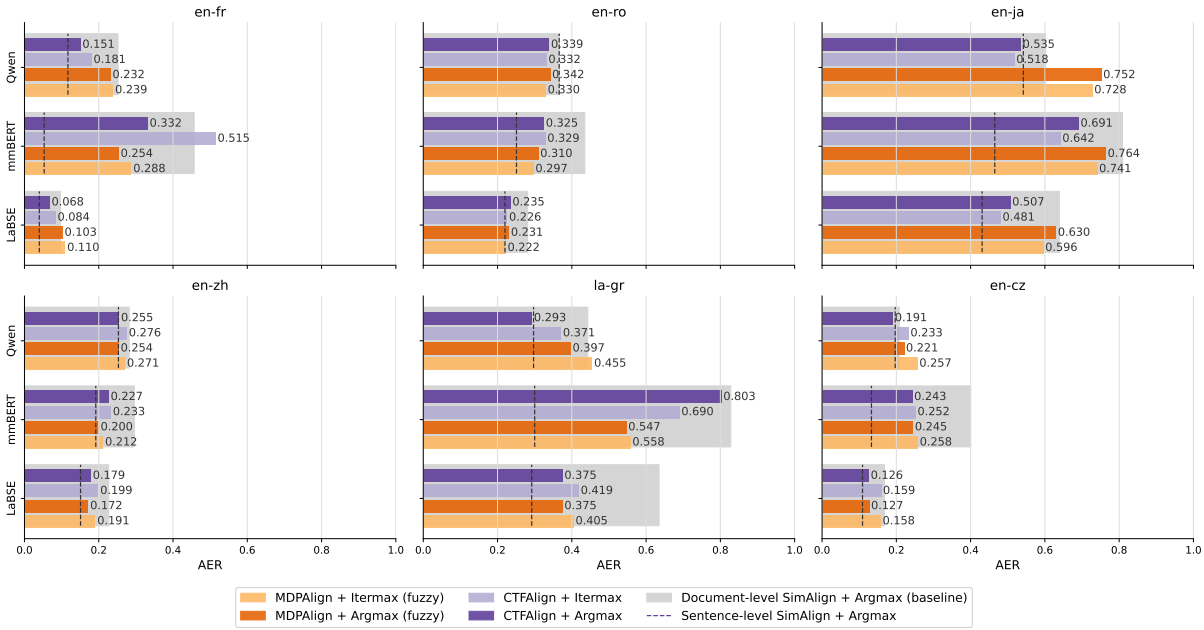


Figure 4: AER (lower is better) on the test set with `Qwen3-Embedding-4B`, `mmBERT-base`, and `LaBSE`. Different constrained word alignment approaches are denoted by different colors, the document-level baseline is shown in gray, and the dashed lines indicate sentence-level SimAlign performance of the given model.

Granularity	en-fr	en-ro	en-ja	en-zh	la-gr	en-cz
Sentences	0.185	0.437	0.628	0.399	0.179	0.344
Documents	0.995	0.968	0.986	0.935	0.996	0.982

Table 1: GPT-5.4-mini word alignment AER on the test set.

CTFAlign in combination with both Argmax and Itermax on the development set. We perform hyperparameter searches over the parameters k and w (Appendix E.2). Based on best development set performance (Appendix E.3), we select MDPAlign *fuzzy* with $k = 150$ and CTFAlign with $w = 8$ for evaluation on the test set.

Evaluation We follow word alignment evaluation from prior work (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Miao et al., 2025) and report alignment error rate (AER) on the test set in Figure 4 for `Qwen3-Embedding-4B`, `mmBERT-base`, and `LaBSE`. In Appendix G, we include F1, Precision, Recall, and corresponding macro averages. Find formalizations of the metrics in Appendix B.2.

5 Results & Discussion

Unconstrained document-level search causes alignment collapse Figure 4 shows that across all models and language pairs, directly applying SimAlign to full documents (gray areas) leads to a clear degradation in alignment quality compared to sentence-level performance (dashed lines). This effect becomes especially pronounced for long documents (en–fr, en–ja, la–gr) with `mmBERT`, where AER values can exceed 0.8 (en–ja, la–gr) at the document level. Manual inspection suggests that many of the resulting spurious alignments involve named entities that are repeated across the document as well as conjunctions, prepositions, and punctuation marks.

LLM prompting does not scale to document-level word alignment Our experiments with `GPT-5.4-mini` reveal that autoregressively generating word alignment labels does not scale to documents (Table 1): while sentence-level performance is competitive on some language pairs (e.g., la–gr, en–fr), AER collapses to near-1.0 across all six pairs once the model is given full documents. These observations suggest that, at least without fine-tuning on document-level alignment data, similarity-matrix-based approaches remain better suited for this task than 1-shot prompting LLMs. This is consistent with Miao et al. (2025), who report that LLM-based word alignment becomes competitive only after supervised fine-tuning.

Structural constraints recover document-level word alignment quality Both of our structural constraints recover most of the performance loss introduced by applying SimAlign directly to

full documents, in several settings closing the sentence- vs. document-level gap entirely. At the individual language-pair level, the gap narrows to within a few thousandths AER: en–ro with LaBSE+MDPAlign+Itermax (0.222 document- vs. 0.220 sentence-level), and en–zh with mmBERT+MDPAlign+Argmax (0.200 vs. 0.192). For Qwen, we even see document-level word alignment surpass sentence-level word alignment for several pairs: en–ro across all configurations, en–ja with CTFAlign, la–gr and en–cz with CTFAlign+Argmax.

MDPAlign is effective when alignments remain close to the diagonal MDPAlign improves over the unconstrained document-level baseline on document pairs whose alignment paths stay close to the global diagonal. However, the fixed diagonal prior cannot accommodate document pairs whose alignments do not follow this pattern. This is most visible on en–ja, where MDPAlign falls under the unconstrained baseline (Qwen+MDPAlign 0.752/0.728 vs. SimAlign 0.602). Visualizations of the en–ja gold labels uncover curved alignment paths that deviate from the main diagonal and are therefore systematically penalized by MDPAlign (Figure 1c).

CTFAlign remains robust across varying alignment structures CTFAlign's performance is comparable to MDPAlign on pairs where the diagonal prior is well-suited (en–ro, en–zh), but CTFAlign's recursive refinement of the search space also adapts to curved or shifted alignment regions (Figure 1d). This leads to the strongest results where flexibility matters: on en–ja with LaBSE CTFAlign+Itermax reaches 0.481 AER (closing more than two thirds of the gap to the sentence-level baseline), on la–gr with Qwen CTFAlign+Argmax meets the sentence-level upper bound (0.293 vs. 0.292), and on en–fr with LaBSE it achieves the strongest document-level result overall (0.068), despite en–fr being the longest document pair in the dataset.

Averaged across all six language pairs, CTFAlign+Argmax is the most robust configuration. It brings down average AER from 0.341 to 0.248 with LaBSE, within 0.041 of the sentence-level upper bound (0.207), and closes the gap between sentence- and document-level word alignment with Qwen *(0.294 vs. 0.295).*

Structural constraints interact with encoder characteristics While CTFAlign achieves the best results with LaBSE and Qwen, with mmBERT it is outperformed on average by MDPAlign (0.437 vs. 0.387 AER for the Argmax variants). Unlike the other two encoders, mmBERT is a general-purpose pretrained encoder that has not been explicitly optimized with a pairwise embedding objective or parallel data. CTFAlign still improves over unconstrained mmBERT, but less than MDPAlign. This suggests that the two structural constraints interact differently with the representations produced by the underlying encoder. Since CTFAlign recursively refines alignment regions, errors introduced at coarse resolutions can propagate to later stages, particularly with Itermax (see Appendix H for visualizations). MDPAlign's fixed diagonal band sidesteps this failure mode, making it a safer choice when the encoder provides a weaker cross-lingual similarity signal—provided the alignment structure remains close to the main diagonal.

Pairwise cross-lingual supervision is associated with alignment quality Across nearly all settings, LaBSE remains the strongest embedding model despite being smaller and older than recent multilingual encoders. One possible explanation is exposure to pairwise cross-lingual supervision: LaBSE, which was explicitly trained on large-scale translation pairs, performs best, while Qwen undergoes contrastive embedding training on large-scale multi-task text pairs, including bitext-mining data. In contrast, mmBERT is pretrained as a general-purpose masked language model without parallel data or an explicit pairwise embedding objective. This observation aligns with prior work showing that LaBSE produces particularly well-structured multilingual representations for cross-lingual similarity tasks (Wang et al., 2022; Dale et al., 2023b; Wastl et al., 2026). While the models differ along several dimensions beyond their training objectives and data, these results suggest that pairwise cross-lingual supervision may help produce alignment-friendly multilingual representations. This further motivates adapting recent multilingual encoders with parallel-data objectives (Dou and Neubig, 2021) specifically for word alignment.[4]

[4]We rule out the usage of gold sentence segmentation as the primary reason for LaBSE's strong performance, since additional experiments with sentence embedding concatenation improve results with mmBERT-base only slightly and still leave them well below LaBSE's (Table 9).

6 Downstream Task Evaluation

Beyond intrinsic alignment evaluation, we test whether our approaches transfer to two downstream alignment-based tasks: translation coverage evaluation (Subsection 6.1) and recognition of semantic differences (RSD; Subsection 6.2). Translation coverage evaluation uses token-level correspondences to identify omissions and additions (hallucinations) in machine translation output (Zhou et al., 2021; Li et al., 2022; Dale et al., 2023a; Huang et al., 2024), while RSD aims to measure semantic difference between tokens of comparable documents (Vamvas and Sennrich, 2023; Wastl et al., 2026; Lozano et al., 2026).

For both tasks we use DiffAlign (Vamvas and Sennrich, 2023) as our baseline. DiffAlign assigns each token a dissimilarity score of $1 - \max_j S_{ij}$ from the subword cosine similarity matrix S. We adapt the structural priors from Section 3 to this scoring setup and use the best-performing long-context encoder from our alignment experiments, `Qwen3-Embedding-4B` (layer 20). Word-level scores are obtained by averaging scores across constituent subword tokens.

6.1 Translation Coverage Evaluation

Data & Evaluation For our experiments on translation coverage evaluation, we use document-level translation outputs from the WMT24 and annotations from WMT24 Metrics Shared Task (Kocmi et al., 2024; Freitag et al., 2024).

The data consists of multi-sentence source texts translated by participating systems. We focus on English–German (en–de) and Japanese–Chinese (ja–zh)[5], which have been annotated by professional translators using the Multidimensional Quality Metrics framework (MQM; Freitag et al., 2021). This framework includes multiple error categories including Accuracy/Omission and Accuracy/Addition, which we consider, while discarding the other categories. We reconstruct the segments into documents and convert the span annotations into binary token-level labels (1 = addition/omission, 0 = everything else). Dataset statistics are shown in Appendix C. For evaluation, we follow Dale et al. (2023a,b) and use ROC AUC.

Results Table 2 shows that use of DiffAlign with structural constraints improves over the unconstrained baseline at segment- and document-level. Consistent with the word alignment results, DiffAlign$_{\text{CTF+Argmax}}$ emerges as the strongest configuration at the document level on both language pairs and both error types (ja–zh: +0.055 (omissions), +0.008 (additions); en–de: +0.036 (omissions), +0.02 (additions)). The segment- to document-level degradation is smaller than in the alignment experiments, likely because the smaller unit here is a segment pair, which spans multiple sentences rather than an individual sentence. MDPAlign remains effective for the shorter and likely more rigidly structured segments, while CTFAlign gains the advantage as document-level structural complexity increases.

	ja-zh		en-de	
Method	**Om.**	**Add.**	**Om.**	**Add.**
SEGMENT-LEVEL				
DiffAlign	0.758	0.769	0.856	0.877
DiffAlign$_{MDP+Argmax}$	0.789	**0.783**	0.867	**0.885**
DiffAlign$_{CTF+Argmax}$	**0.801**	0.779	**0.869**	0.882
DiffAlign$_{CTF+Itermax}$	0.781	0.776	0.862	0.880
DOCUMENT-LEVEL				
DiffAlign	0.733	0.763	0.831	0.843
DiffAlign$_{MDP+Argmax}$	0.762	0.762	0.857	0.854
DiffAlign$_{CTF+Argmax}$	**0.788**	**0.771**	**0.867**	**0.863**
DiffAlign$_{CTF+Itermax}$	0.770	0.769	0.853	0.854

Table 2: Token-level ROC AUC for translation coverage evaluation with `Qwen3-Embedding-4B`.

Method	**en–de**	**en–fr**	**en–it**
DiffAlign	0.286	0.156	0.318
DiffAlign$_{MDP+Argmax}$	0.312	0.158	0.312
DiffAlign$_{CTF+Argmax}$	**0.330**	**0.189**	**0.353**
DiffAlign$_{CTF+Itermax}$	0.319	0.172	0.344

Table 3: Test set Spearman ρ for SwissGov-RSD with `Qwen3-Embedding-4B`.

6.2 Recognition of Semantic Differences

Data & Evaluation We evaluate on SwissGov-RSD (Wastl et al., 2026), a human-annotated benchmark for token-level recognition of semantic differences between related cross-lingual documents drawn from Swiss government sources. The benchmark covers three language pairs English–German (en–de), English–French (en–fr), English–Italian (en–it). We follow their evaluation protocol and report Spearman correlations between predicted and gold token-level scores.

[5] We exclude English–Spanish because its annotations differ from the remaining language pairs.

Results Table 3 shows that the same pattern holds for recognition of semantic differences: search-space restriction improves over the DiffAlign baseline across all three language pairs (+0.044 for en–de, +0.033 for en–fr, +0.035 for en–it), with $\text{DiffAlign}_{\text{CTF+Argmax}}$ yielding the highest correlations with gold labels and establishing new state-of-the-art results on SwissGov-RSD. Notably, MDPAlign offers little improvement across languages on this task, even underperforming the baseline on en–it. This suggests that CTFAlign's recursive search-space restriction is particularly well-suited to this setting of imperfectly parallel documents, containing subtle semantic shifts as well as entirely omitted or added paragraphs, with gold annotations obtained directly at the document level.

Across both downstream tasks, CTFAlign+Argmax remains the best-performing configuration, suggesting that its gains transfer well to realistic alignment-based downstream applications.

7 Conclusion

We introduce two lightweight, training-free approaches for document-level word alignment: MDPAlign, which imposes a global positional prior, and CTFAlign, which recursively restricts alignment to semantically plausible regions. Across six language pairs, both methods substantially reduce the degradation observed when moving from sentences to full documents, with CTFAlign leading to the most robust performance across diverse document structures and language pairs.

These improvements transfer beyond intrinsic alignment evaluation. We show that CTFAlign improves the downstream tasks of translation coverage evaluation and semantic difference recognition, demonstrating that document-level word alignment can benefit cross-lingual document comparison tasks without sentence segmentation, additional supervision, or model retraining.

Limitations

Sentence-level nature of the word alignment dataset Word alignment gold data was tailored to sentences and had to be reconstructed to documents. This has the following consequences:

- Document structures remain largely monotonic, they may not contain reordering, insertions, or deletions.
- Sentence-level annotations may fail to capture document-level phenomena such as long-range reordering, cross-sentence dependencies, discourse-driven translation shifts, or omissions and additions spanning multiple sentences.
- Concatenation of documents led to a small number of document pairs to evaluate on, e.g., the en–fr and la–gr test sets each consist of only a single document pair.

These limitations further motivate the inclusion of downstream document-level evaluations. In particular, while the WMT-based data was reconstructed from segment-level annotations, annotators were provided with broader contextual information during annotation. Moreover, SwissGov-RSD was annotated directly at the document level and therefore captures document-level semantic phenomena more naturally.

Dependency on encoder representations Our approaches operate directly on token similarity matrices and therefore depend on the representations produced by the underlying multilingual encoder. When cross-lingual token similarities are insufficiently distinctive, structural constraints cannot fully compensate, and iterative refinement strategies such as CTFAlign may propagate errors introduced at earlier stages. At the same time, this limitation suggests a promising direction for future work: long-context multilingual encoders could be further optimized to produce representations better suited for fine-grained cross-lingual alignment.

Ethical Considerations

No specific ethical risks have been identified for this work. The artifacts used in this work, except for the proprietary LLM, have open licenses and due to their origin, the datasets are unlikely to contain personally identifiable information or offensive content.

Acknowledgments

This work was funded by the Swiss National Science Foundation (project InvestigaDiff; no. 10000503). Additionally, we would like to thank Marius Huber for helpful feedback.

References

Yuki Arase, Han Bao, and Sho Yokoi. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3966–3986, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte Miguel Alves, Andre Martins, Ayoub Hammal, Caio Corro, Celine Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El Haddad, Manuel Faysse, Maxime Peyrard, Nuno M Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. EuroBERT: Scaling multilingual encoders for european languages. In *Second Conference on Language Modeling*.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Yang Chen, Chao Jiang, Alan Ritter, and Wei Xu. 2023. Frustratingly easy label projection for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5775–5796, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Ido Dagan, Kenneth Church, and Willian Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Very Large Corpora: Academic and Industrial Perspectives*.

David Dale, Elena Voita, Loic Barrault, and Marta R Costa-jussà. 2023a. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–50, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Dale, Elena Voita, Janice Lam, Prangthip Hansanti, Christophe Ropers, Elahe Kalbassi, Cynthia Gao, Loïc Barrault, and Marta R Costa-jussà. 2023b. HalOmi: A manually annotated benchmark for multilingual hallucination and omission detection in machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 638–653. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North*, pages 4171–4186, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 118–124.

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

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023. SemEval-2023 task 2: Fine-grained multilingual named entity recognition (MultiCoNER 2). In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2247–2265, Stroudsburg, PA, USA. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Trans. Assoc. Comput. Linguist.*, 9:1460–1474.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference*

on Machine Translation, pages 47–81, Stroudsburg, PA, USA. Association for Computational Linguistics.

William A Gale and Kenneth Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Iker García-Ferrero, Rodrigo Agerri, and German Rigau. 2022. Model and data transfer for cross-lingual sequence labelling in zero-resource settings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6403–6416, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chenyang Huang, Abbas Ghaddar, Ivan Kobyzev, Mehdi Rezagholizadeh, Osmar Zaiane, and Boxing Chen. 2024. OTTAWA: Optimal TransporT adaptive word aligner for hallucination and omission translation errors detection. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6322–6334, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 others. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiang Kong, Zhaopeng Tu, Shuming Shi, Eduard Hovy, and Tong Zhang. 2019. Neural machine translation with adequacy-oriented learning. *Proc. Conf. AAAI Artif. Intell.*, 33(01):6618–6625.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. TransLLaMa: LLM-based simultaneous translation system. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 461–476, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gaetan Latouche, Marc-André Carbonneau, and Benjamin Swanson. 2024. BinaryAlign: Word alignment as binary sequence labeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10277–10288, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv [cs.CL]*.

Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. *Proc. Conf. AAAI Artif. Intell.*, 29(1).

Diego Miguel Lozano, Daryna Dementieva, and Alexander Fraser. 2026. Explainable semantic textual similarity via dissimilar span detection. In *Proceedings of the Language Resources and Evaluation Conference*, pages 2190–2210. European Language Resources Association (ELRA).

David Mareček. 2011. Automatic alignment of tectogrammatical trees from czech-english parallel corpus.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. MmBERT: A modern multilingual encoder with annealed language learning. *arXiv [cs.CL]*.

Zhongtao Miao, Qiyu Wu, Masaaki Nagata, and Yoshimasa Tsuruoka. 2025. Improving word alignment using semi-supervised learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19871–19888, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhongtao Miao, Qiyu Wu, Kaiyan Zhao, Zilong Wu, and Yoshimasa Tsuruoka. 2024. Enhancing cross-lingual sentence embedding for low-resource languages with word alignment. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3225–3236, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10.

Graham Neubig. 2011. The kyoto free translation task. http://www.phontron.com/kftt.

Slav Petrov. 2011. *Coarse-to-fine natural language processing*, 2012 edition. Theory and Applications of Natural Language Processing. Springer, Berlin, Germany.

Slav Petrov, Aria Haghighi, and Dan Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, pages 108–116, Morristown, NJ, USA. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2023. Towards unsupervised recognition of token-level semantic differences in related documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13543–13552, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hao Wang, Linlong Xu, Heng Liu, Yangyang Liu, Xiaohu Zhao, Bo Zeng, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. Marco large translation model at WMT2025: Transforming translation capability in LLMs via quality-aware training and decoding. In *Proceedings of the Tenth Conference on Machine Translation*, pages 587–593, Stroudsburg, PA, USA. Association for Computational Linguistics.

Weikang Wang, Guanhua Chen, Hanqing Wang, Yue Han, and Yun Chen. 2022. Multilingual sentence transformer as a multilingual word aligner. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2952–2963, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michelle Wastl, Jannis Vamvas, and Rico Sennrich. 2026. SwissGov-RSD: A human-annotated, cross-lingual benchmark for token-level recognition of semantic differences between related documents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31134–31163.

Qiyu Wu, Masaaki Nagata, Zhongtao Miao, and Yoshimasa Tsuruoka. 2024. Word alignment as preference for machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3223–3239, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qiyu Wu, Masaaki Nagata, and Yoshimasa Tsuruoka. 2023. WSPAlign: Word alignment pre-training via large-scale weakly supervised span prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11084–11099, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tariq Yousef, Chiara Palladino, Farnoosh Shamsian, Anise D'orange Ferreira, and Michel Ferreira dos Reis. 2022. An automatic model and gold standard for translation alignment of ancient greek. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5894–5905.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. MGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv [cs.CL]*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Stroudsburg, PA, USA. Association for Computational Linguistics.

A CTFAlign Reference Implementation

```
def ctfalign(S, align, width):
    """
    S: m x n similarity matrix (source tokens x target tokens)
    align: base word alignment function, e.g. SimAlign
           argmax/itermax, returns a set of (row, col)
           index pairs
    width: buffer width w (in coarse blocks) kept around
           each aligned block
    """
    m, n = S.shape
    block_h = max(m // 2, 1)   # start from a 2x2 grid
    block_w = max(n // 2, 1)

    while block_h >= 1 and block_w >= 1:
        # 1. Coarsen: average-pool S into blocks
        coarse = average_pool(S, block_h, block_w)  # R x C
        R, C = coarse.shape

        # 2. Align on the coarse (block-level) matrix
        coarse_alignment = align(coarse)  # set of (i, j)

        # 3. Stop once blocks are single tokens
        if block_h == 1 and block_w == 1:
            return coarse_alignment

        # 4. Mark retained blocks: aligned block +- width
        mask = zeros(R, C)
        for (i, j) in coarse_alignment:
            r0, r1 = max(i - width, 0), min(i + width + 1, R)
            c0, c1 = max(j - width, 0), min(j + width + 1, C)
            mask[r0:r1, c0:c1] = 1

        # 5. Recovery: empty rows/cols get intersecting
        #    cells recovered, also with +- width buffer
        empty_rows = [i for i in range(R)
                      if mask[i, :].sum() == 0]
        empty_cols = [j for j in range(C)
                      if mask[:, j].sum() == 0]
        for i in empty_rows:
            for j in empty_cols:
                r0, r1 = max(i - width, 0), min(i + width + 1, R)
                c0, c1 = max(j - width, 0), min(j + width + 1, C)
                mask[r0:r1, c0:c1] = 1

        # 6. Zero out S wherever the mask is 0
        S = S * upsample(mask, block_h, block_w)

        # 7. Refine: halve block size, clamp at 1
        block_h = max(block_h // 2, 1)
        block_w = max(block_w // 2, 1)
```

Figure 5: Simplified Python code for CTFAlign. Tensor-level details (average pooling and mask upsampling implemented via PyTorch's `avg_pool2d`/`interpolate`) are abstracted for readability.

B Evaluation Details

B.1 Dataset Statistics

Table 4 summarizes the statistics of the reconstructed document-level word alignment datasets used in our experiments. The datasets vary considerably in document length, number of sentences, and split sizes, covering both short texts (en–zh) and very long documents (en–fr, la–gr).

Lang	Docs		Sents		Mean len	
	dev	test	dev	test	doc	sent
en-fr	1	1	37	447	22408	92
en-ro	1	10	17	248	2901	119
en-ja	8	7	653	582	5516	66
en-zh	47	52	450	450	646	70
la-gr	1	1	25	75	5837	116
en-cz	–	33	–	1636	6318	126

Table 4: Word alignment dataset statistics. Lengths are target-side (*text_b*) character counts; only means shown for space. – indicates no data for that split.

B.2 Metric formulas

Gold alignments usually consist of two sets: sure alignments S and possible alignments P.

$$\text{Precision}(\mathcal{A}, \mathcal{P}) = \frac{|\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}|} \tag{6}$$

$$\text{Recall}(\mathcal{A}, \mathcal{S}) = \frac{|\mathcal{A} \cap \mathcal{S}|}{|\mathcal{S}|} \tag{7}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

$$\text{AER}(\mathcal{A}, \mathcal{S}, \mathcal{P}) = 1 - \frac{|\mathcal{A} \cap \mathcal{S}| + |\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}| + |\mathcal{S}|} \tag{9}$$

Where $\mathcal{P}$ is defined as $S \cup P$. Metrics are reported per language pair.

C Details on Omission and Addition Errors in WMT24 System Outputs

We evaluate our experiments for translation coverage evaluation with human annotated WMT24 system translations (Kocmi et al., 2024; Freitag et al., 2024). In Table 5, we provide an overview over the total number of documents included in the dataset for language pair, the number of documents, which include at least on addition or omission, information about length and total error percentage.

Lang	Docs		Sents		Mean len		Err %	
	tot.	err.	tot.	err.	doc	sent	add.	omi.
en-de	3230	284	9232	320	724	253	0.3	0.2
ja-zh	2940	554	8385	617	264	92	1.0	1.1

Table 5: WMT 2024 MQM dataset statistics. Doc/sent lengths are target-side character counts; only means shown. *err.* columns count documents/sentences containing at least one omission or addition. *add.* and *omi.* report the fraction (%) of non-punctuation characters labeled as additions or omissions.

D LLM Baseline

The 1-shot prompt is shown in Figure 6. The example in the prompt is a random text pair with gold labels of the corresponding language pair from development set. If the language pair does not have a development set, the example is from en–fr.

E Experiments on the Development Set

E.1 Model selection and granularity gap

Table 6 reports pilot experiment results on the development set across a larger set of embedding models (including `XLM-R` (large; (Conneau et al., 2020) and `mBERT` (Devlin et al., 2019) at the sentence-level as well as `EuroBERT-610m` (Boizard et al., 2025) and `Qwen3-Embedding-0.6` (Zhang et al., 2025) for both granularities and both SimAlign variants as macro-averages across languages. The results reveal a degradation when moving from sentence- to document-level alignment. They also motivate our model selection for the main experiments: `LaBSE`, `Qwen-Embedding-4B`, and `mmBERT-base` emerge as the strongest multilingual embedding models overall.

E.2 Hyperparameter selection

Model layer Previous work indicates that word alignment performance varies with embedding model layers and that the default last layer oftentimes underperforms in comparison to middle layers (Zhang et al., 2019; Jalili Sabet et al., 2020). Hence, we search for the best layer for each language pair and granularity.

The layer search reported in Figure 7 reveals that the layer selection becomes more important at the document level with differences of over 25 F1 points between the best layer and the last. Furthermore, document pairs tend to benefit from later layers than sentence pairs for word alignment. For all further reported results we use the best performing

```
[SYSTEM]
You are a helpful AI assistant for word alignment.

[USER]
Please give me the word alignment information between the following two texts. The parallel texts:
each_0 of_1 them_2 is_3 very_4 complex_5 ,_6 but_7 the_8 link_9 between_10 the_11 two_12 is_13
even_14 more_15 complex_16 which_17 makes_18 the_19 whole_20 situation_21 for_22 most_23 people_24
understandably_25 confusing_26 ._27

chacun_0 en_1 lui_2 -_3 même_4 est_5 très_6 complexe_7 et_8 le_9 lien_10 entre_11 les_12 deux_13
le_14 est_15 encore_16 davantage_17 de_18 sorte_19 que_20 pour_21 beaucoup_22 la_23 situation_24
présente_25 est_26 confuse_27 ._28

[ASSISTANT]
{"alignments": [{"source_idx": 0, "target_idxs": [0]}, {"source_idx": 3, "target_idxs": [5]},
{"source_idx": 4, "target_idxs": [6]}, {"source_idx": 5, "target_idxs": [7]}, {"source_idx": 8,
"target_idxs": [9]}, {"source_idx": 9, "target_idxs": [10]}, {"source_idx": 10, "target_idxs":
[11]}, {"source_idx": 11, "target_idxs": [12]}, {"source_idx": 12, "target_idxs": [13]},
{"source_idx": 13, "target_idxs": [15]}, {"source_idx": 19, "target_idxs": [23]}, {"source_idx":
21, "target_idxs": [24]}, {"source_idx": 22, "target_idxs": [21]}, {"source_idx": 27, "target_idxs":
[28]}]}

[USER]
Please give me the word alignment information between the following two texts. The parallel texts:
[...]

[ASSISTANT]
{"alignments": [...]}
```

Figure 6: 1-shot prompt example used for the LLM baseline based on the "full mode" prompt introduced by Miao et al. (2025).

layers per language pair and granularity (marked with a star in Figure 7).

Width hyperparameters k and w Both hyperparameters first increase performance with a growing value for their corresponding algorithm but do not substantially change the performance once a plateau is reached (as shown in Figure 8). Hence, we decide to fix the bandwidth parameter k and w at the first value that reaches that plateau on average across language pairs and models (Table 8). For MDPAlign (strict) a narrow bandwidth of $k = 50$ is optimal, where the maximum value of $k = 150$ for MDPAlign (fuzzy) and $w = 8$ for CTFAlign performs best. For all further experiments, we fix the values according to Table 8.

E.3 Results & Discussion

Development set results are presented in Table 9. Directly applying SimAlign at document level leads to degradation compared to sentence-level alignment with both Argmax and Itermax. Since both Armgax and Itermax perform comparably at the document-level, we continue to use only the argmax variants as a baseline in all further experiments. Furthermore, experiments with increasing the iteration count of Itermax leads to further degradation of the performance while also increasing runtime. The document-level degradation is particularly severe for recall, indicating that unconstrained similarity matrices contain many competing high-scoring similarities that cause greedy alignment decisions to miss correct correspondences. Both MDPAlign and CTFAlign improve recall by restricting the search space and suppressing spurious long-range matches. Strict MDPAlign has the overall lowest performance, hence we do not proceed to evaluate it on the test set in the main paper.

F SwissGov-RSD Development Set Results

Table 7 shows development set results on SwissGov-RSD using `Qwen3-Embedding-4B`. Across all language pairs, applying structural constraints consistently improves over the unconstrained DiffAlign baseline, indicating that restricting the search space is beneficial for document-level semantic difference detection.

As is the case in test experiments, our approaches outperform the baseline across all lan-

Model / Method	P	R	F1	AER
SENTENCE-LEVEL				
XLM-R-large				
SimAlign + Argmax	0.777	0.635	0.692	0.308
SimAlign + Itermax	0.689	0.692	0.685	0.317
mBERT				
SimAlign + Argmax	0.816	0.625	0.701	0.297
SimAlign + Itermax	0.711	0.711	0.707	0.295
mmBERT				
SimAlign + Argmax	0.844	0.710	0.763	0.236
SimAlign + Itermax	0.733	0.791	0.755	0.248
EuroBERT-610m				
SimAlign + Argmax	0.671	0.516	0.573	0.426
SimAlign + Itermax	0.545	0.596	0.563	0.442
LaBSE				
SimAlign + Argmax	0.858	0.727	0.781	0.218
SimAlign + Itermax	0.753	0.799	0.772	0.232
Qwen-0.6B				
SimAlign + Argmax	0.780	0.601	0.670	0.329
SimAlign + Itermax	0.672	0.681	0.669	0.334
Qwen-4B				
SimAlign + Argmax	0.807	0.638	0.702	0.297
SimAlign + Itermax	0.699	0.682	0.685	0.316
DOCUMENT-LEVEL				
mmBERT				
SimAlign + Argmax	0.706	0.453	0.532	0.467
SimAlign + Itermax	0.616	0.531	0.553	0.449
EuroBERT-610m				
SimAlign + Argmax	0.621	0.358	0.427	0.572
SimAlign + Itermax	0.508	0.414	0.433	0.571
LaBSE-concatenated-sentences				
SimAlign + Argmax	**0.771**	0.495	0.594	0.401
SimAlign + Itermax	0.688	0.574	0.617	0.378
Qwen-0.6B				
SimAlign + Argmax	0.740	0.496	0.580	0.419
SimAlign + Itermax	0.640	0.565	0.588	0.413
Qwen-4B				
SimAlign + Argmax	0.768	0.568	**0.643**	**0.355**
SimAlign + Itermax	0.657	**0.642**	0.639	0.363

Table 6: Macro-average pilot word alignment development results across language pairs. Best sentence-level results are underlined and best document-level results are **bolded**. For AER lower is better.

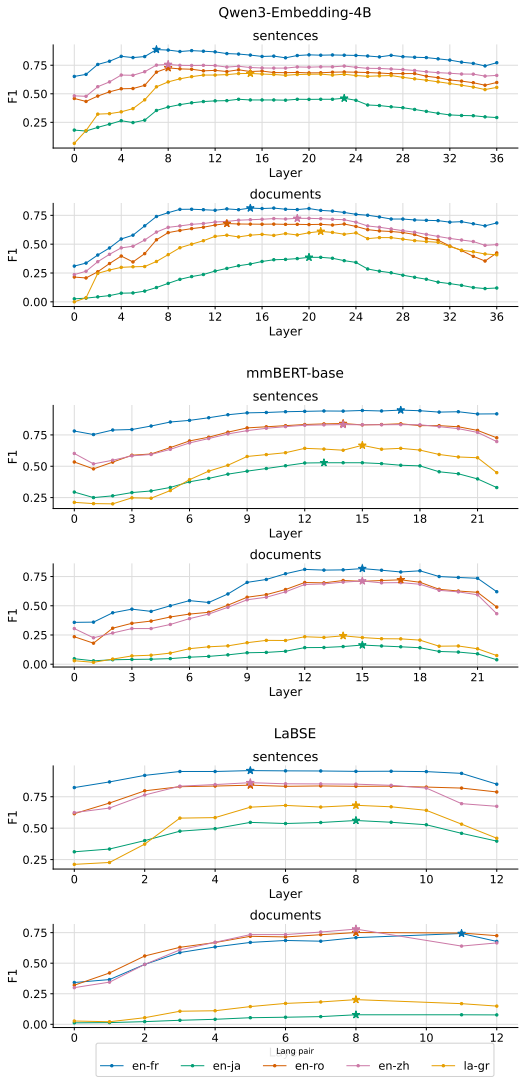


Figure 7: F1 word alignment results across layers for different language pairs with Argmax on the development set for both granularities for `Qwen3-Embedding-4B`, `mmBERT-base`, and `LaBSE-concatenated-sentences`.

guage pairs while $\text{DiffAlign}_{\text{CTF+Argmax}}$ achieves the strongest overall performance. The gains are especially pronounced at layer 20, which is on average across languages the best layer for document-level word alignment according to our layer search described in Appendix E.2.

The gap between the final layer and layer 20 is large across all methods, aligning with prior observations (Zhang et al., 2019; Jalili Sabet et al., 2020) as well as ours that the final layers of multilingual embedding models are often suboptimal for cross-lingual similarity tasks.

G Main Results with Precision, Recall and F1

In Table 10, we report precision, recall, and F1 in addition to AER to complement the results shown in Figure 4. The non-AER metrics reveal a consistent precision–recall tradeoff between alignment strategies. Argmax-based variants favor precision and more often achieve the strongest F1 scores, whereas Itermax generally improves recall by recovering additional alignments at the cost of precision. This pattern holds across both MDPAlign and CTFAlign. Neither Itermax variant dominates recall consistently: CTFAlign+Itermax achieves higher average recall with `LaBSE` and `Qwen`, while MDPAlign+Itermax does so with `mmBERT`. Overall, the results show that the choice between Argmax and Itermax primarily controls the precision–recall balance, while the structural constraint determines how successfully this tradeoff is managed for a given encoder.

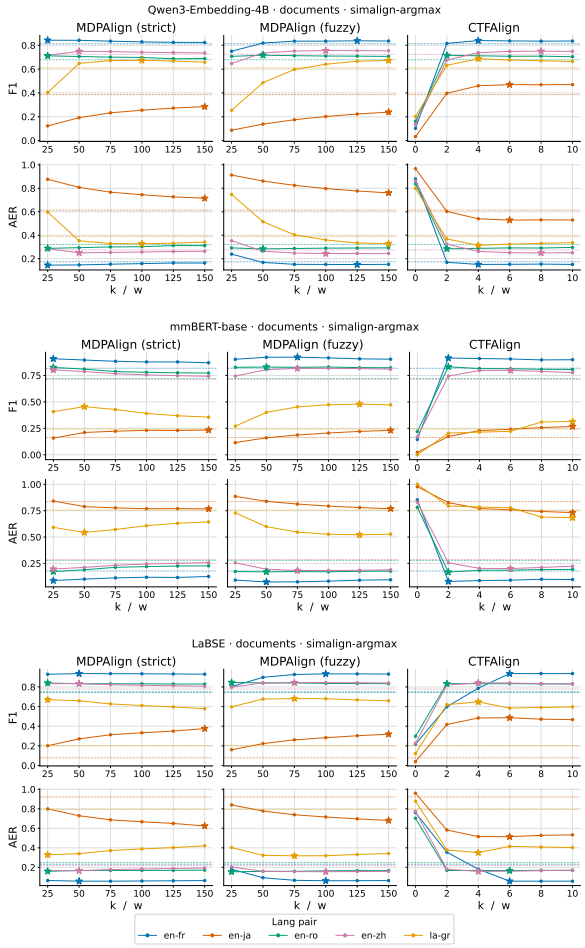


Figure 8: AER (lower is better) and F1 (higher is better) results for different k/w values for `Qwen3-Embedding-4B`, `mmBERT-base`, and `LaBSE-concatenated-sentences`.

H `mmBERT` Failure Case

In Figure 9, we visualize the failure case on the en–fr test set example with `mmBERT` embeddings and CTFAlign+Itermax. The failure case illustrates that when combining CTFAlign with Itermax on a noisy similarity matrix the approach is too lenient and results in spurious alignment regions, which are likely the result of falsely aligned coarse similarity blocks at a lower resolution.

Method	en–de	en–fr	en–it
LAST LAYER			
DiffAlign	0.186	0.098	0.206
DiffAlign$_{\text{MDP+Argmax}}$	0.227	0.110	0.228
DiffAlign$_{\text{CTF+Itermax}}$	0.262	0.140	0.254
LAYER 20			
DiffAlign	0.286	0.156	0.318
DiffAlign$_{\text{MDP+Argmax}}$	0.301	0.168	0.309
DiffAlign$_{\text{CTF+Argmax}}$	**0.346**	**0.208**	**0.365**
DiffAlign$_{\text{CTF+Itermax}}$	0.332	0.199	0.351

Table 7: SwissGov-RSD development set Spearman ρ for DiffAlign variants using `Qwen3-Embedding-4B` across en–de, en–fr, and en–it. Bold indicates the best result per language pair.

Approach	k/w	F1	AER
CTFAlign	0	0.140	0.858
	2	0.626	0.369
	4	0.666	0.331
	6	0.673	0.325
	8	**0.677**	**0.321**
	10	**0.677**	**0.322**
MDPAlign (strict)	25	0.625	0.374
	50	**0.655**	**0.343**
	75	0.654	0.345
	100	0.651	0.348
	125	0.648	0.351
	150	0.646	0.353
MDPAlign (fuzzy)	25	0.567	0.431
	50	0.633	0.365
	75	0.656	0.343
	100	0.664	0.334
	125	0.668	0.331
	150	**0.669**	**0.330**

Table 8: Average dev-set F1 and AER (lower is better) across `LaBSE`, `mmBERT`, and `Qwen` for each hyperparameter value. Approaches were combined with Argmax. Best per approach and column in **bold**.

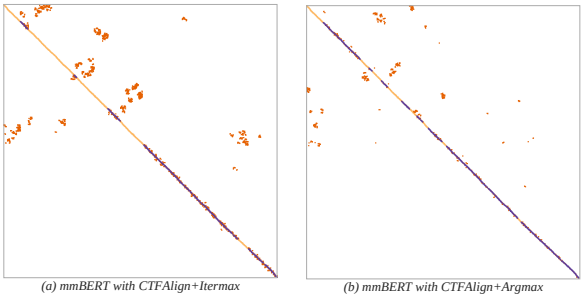


Figure 9: (a) CTFAlign + Itermax with `mmBERT` failure cases on en–fr in comparison to partially recovered alternative (b) CTFAlign + Argmax. Colors denote correct alignments, false negatives, false positives.

Model	en-fr				en-ro				en-ja				en-zh				la-gr				Avg			
	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER
SENTENCE-LEVEL																								
GPT-5.4-mini	0.820	0.799	0.809	0.187	0.632	0.592	0.611	0.389	0.255	0.576	0.354	0.646	0.547	0.669	0.602	0.398	0.762	0.849	0.803	0.198	0.603	0.697	0.636	0.364
LaBSE																								
SimAlign + Argmax	0.966	0.950	0.958	0.040	0.971	0.743	0.842	0.158	0.761	0.444	0.561	0.439	0.892	0.834	0.862	0.138	0.702	0.664	0.682	0.317	0.858	0.727	0.781	0.218
SimAlign + Itermax	0.886	0.979	0.930	0.083	0.874	0.834	0.854	0.146	0.652	0.532	0.586	0.414	0.777	0.896	0.832	0.168	0.604	0.753	0.670	0.331	0.759	0.799	0.774	0.228
mmBERT																								
SimAlign + Argmax	0.968	0.929	0.948	0.048	0.960	0.749	0.841	0.159	0.750	0.408	0.528	0.471	0.882	0.788	0.832	0.167	0.660	0.674	0.667	0.333	0.844	0.710	0.763	0.236
SimAlign + Itermax	0.868	0.967	0.915	0.098	0.875	0.842	0.858	0.142	0.635	0.502	0.561	0.439	0.746	0.883	0.809	0.192	0.539	0.760	0.631	0.371	0.733	0.791	0.755	0.248
Qwen																								
SimAlign + Argmax	0.943	0.837	0.887	0.104	0.955	0.587	0.727	0.273	0.663	0.353	0.461	0.540	0.837	0.691	0.757	0.243	0.638	0.723	0.678	0.323	0.807	0.638	0.702	0.297
SimAlign + Itermax	0.824	0.891	0.856	0.151	0.828	0.648	0.727	0.273	0.548	0.434	0.484	0.515	0.737	0.738	0.737	0.262	0.557	0.699	0.620	0.381	0.699	0.682	0.685	0.316
DOCUMENT-LEVEL																								
GPT-5.4-mini	0.072	0.068	0.070	0.929	0.024	0.022	0.023	0.977	0.010	0.010	0.010	0.990	0.065	0.079	0.071	0.929	0.095	0.114	0.104	0.897	0.053	0.059	0.056	0.944
LaBSE																								
SimAlign + Argmax	0.933	0.533	0.678	0.297	0.950	0.621	0.751	0.249	0.632	0.236	0.344	0.656	0.858	0.714	0.779	0.220	0.480	0.370	0.418	0.581	0.771	0.495	0.594	0.401
MDPAlign + Argmax (strict)	**0.969**	0.905	**0.936**	**0.058**	**0.976**	0.727	0.833	0.167	0.451	0.194	0.271	0.729	0.868	0.802	0.834	0.166	**0.685**	0.635	0.659	0.340	0.790	0.653	0.707	0.292
MDPAlign + Argmax (fuzzy)	0.958	0.905	0.931	0.064	0.971	0.730	0.833	0.166	0.514	0.231	0.319	0.681	**0.871**	0.812	**0.840**	**0.159**	0.684	0.635	0.659	0.341	0.800	0.663	0.716	0.282
CTFAlign + Argmax	**0.969**	0.905	**0.936**	**0.058**	0.955	0.736	0.831	0.169	**0.722**	0.351	0.472	0.528	0.867	0.804	0.834	0.166	0.645	0.546	0.591	0.408	**0.832**	0.668	**0.733**	**0.266**
MDPAlign + Itermax (strict)	0.910	0.929	0.919	0.083	0.868	0.829	0.848	0.152	0.363	0.238	0.288	0.713	0.758	0.878	0.814	0.187	0.580	0.721	0.643	0.359	0.696	0.719	0.702	0.299
MDPAlign + Itermax (fuzzy)	0.906	0.935	0.920	0.083	0.870	**0.831**	**0.850**	**0.150**	0.424	0.289	0.344	0.656	0.760	**0.883**	0.817	0.184	0.583	0.721	0.645	0.357	0.709	**0.732**	0.715	0.286
CTFAlign + Itermax	0.909	0.929	0.919	0.084	0.872	0.829	**0.850**	**0.150**	0.608	**0.462**	**0.525**	**0.475**	0.757	0.874	0.811	0.189	0.512	0.506	0.509	0.491	0.732	0.720	0.723	0.278
mmBERT																								
SimAlign + Argmax	0.851	0.787	0.818	0.178	0.958	0.579	0.722	0.278	0.533	0.097	0.164	0.836	0.862	0.606	0.712	0.288	0.324	0.195	0.243	0.756	0.706	0.453	0.532	0.467
MDPAlign + Argmax (strict)	0.917	0.876	0.896	0.101	0.953	0.705	0.810	0.190	0.372	0.147	0.211	0.789	0.857	0.728	0.787	0.212	0.467	0.444	0.455	0.544	0.713	0.580	0.632	0.367
MDPAlign + Argmax (fuzzy)	0.909	0.899	0.904	0.095	0.961	0.719	0.823	0.177	0.401	0.162	0.231	0.769	0.867	0.759	0.809	0.190	0.486	0.459	0.472	0.528	0.725	0.600	0.648	0.352
CTFAlign + Argmax	0.916	0.879	0.897	0.100	0.958	0.699	0.808	0.192	0.495	0.174	0.257	0.743	0.855	0.731	0.788	0.212	0.345	0.281	0.310	0.690	0.714	0.553	0.612	0.387
MDPAlign + Itermax (strict)	0.776	0.947	0.853	0.165	0.874	0.798	0.834	0.166	0.296	0.187	0.229	0.771	0.743	0.846	0.791	0.209	0.364	0.548	0.437	0.564	0.611	0.665	0.629	0.375
MDPAlign + Itermax (fuzzy)	0.771	**0.956**	0.854	0.166	0.881	0.811	0.845	0.156	0.323	0.206	0.252	0.748	0.747	0.863	0.801	0.200	0.380	0.565	0.454	0.546	0.620	0.680	0.641	0.363
CTFAlign + Itermax	0.782	0.947	0.857	0.159	0.866	0.787	0.825	0.176	0.411	0.225	0.291	0.709	0.745	0.836	0.788	0.212	0.251	0.341	0.289	0.711	0.611	0.627	0.610	0.393
mmBERT-concatenated-sentences																								
SimAlign + Argmax	0.894	0.781	0.834	0.159	0.948	0.563	0.706	0.294	0.656	0.140	0.231	0.770	0.868	0.568	0.687	0.313	0.439	0.299	0.356	0.644	0.761	0.470	0.563	0.436
MDPAlign + Argmax (strict)	0.947	0.891	0.918	0.077	0.967	0.736	0.836	0.164	0.296	0.118	0.169	0.832	0.870	0.756	0.809	0.191	0.506	0.521	0.513	0.487	0.717	0.604	0.649	0.350
MDPAlign + Argmax (fuzzy)	0.946	0.899	0.922	0.074	0.973	0.734	0.837	0.163	0.437	0.173	0.248	0.752	0.884	0.756	0.815	0.185	0.601	0.607	0.604	0.396	0.768	0.634	0.685	0.314
CTFAlign + Argmax	0.937	0.858	0.896	0.097	0.973	0.714	0.824	0.176	0.524	0.186	0.275	0.726	0.867	0.703	0.776	0.223	0.584	0.558	0.571	0.429	0.777	0.604	0.668	0.330
MDPAlign + Itermax (strict)	0.838	0.947	0.889	0.124	0.877	0.831	0.853	0.147	0.227	0.143	0.175	0.825	0.742	0.853	0.794	0.207	0.395	0.590	0.473	0.528	0.616	0.673	0.637	0.366
MDPAlign + Itermax (fuzzy)	0.833	0.953	0.889	0.124	0.892	0.827	0.858	0.142	0.359	0.219	0.272	0.728	0.768	0.864	0.813	0.187	0.486	0.716	0.579	0.422	0.668	0.716	0.682	0.321
CTFAlign + Itermax	0.818	0.932	0.871	0.141	0.889	0.801	0.843	0.157	0.474	0.240	0.319	0.682	0.759	0.819	0.788	0.212	0.444	0.610	0.514	0.487	0.677	0.680	0.667	0.336
Qwen																								
SimAlign + Argmax	0.894	0.746	0.813	0.174	0.901	0.545	0.679	0.321	0.626	0.279	0.386	0.614	0.812	0.654	0.724	0.275	0.608	0.615	0.611	0.389	0.768	0.568	0.643	0.355
MDPAlign + Argmax (strict)	0.905	0.787	0.842	0.147	0.863	0.597	0.706	0.294	0.325	0.136	0.192	0.808	0.799	0.706	0.750	0.250	0.604	0.699	0.648	0.353	0.699	0.585	0.628	0.370
MDPAlign + Argmax (fuzzy)	0.900	0.781	0.836	0.152	0.868	0.597	0.707	0.292	0.388	0.173	0.239	0.761	0.801	0.713	0.754	0.245	0.623	0.731	**0.673**	**0.329**	0.716	0.599	0.642	0.356
CTFAlign + Argmax	0.896	0.781	0.835	0.154	0.868	0.599	0.709	0.291	0.639	0.370	0.469	0.531	0.798	0.709	0.751	0.249	0.622	0.726	0.670	0.331	0.765	0.637	0.687	0.311
MDPAlign + Itermax (strict)	0.786	0.840	0.812	0.194	0.765	0.676	0.718	0.282	0.249	0.165	0.198	0.802	0.671	0.808	0.733	0.268	0.457	0.743	0.566	0.437	0.586	0.646	0.605	0.397
MDPAlign + Itermax (fuzzy)	0.777	0.840	0.807	0.200	0.764	0.674	0.716	0.284	0.313	0.218	0.257	0.743	0.671	0.815	0.736	0.265	0.481	0.778	0.594	0.409	0.601	0.665	0.622	0.380
CTFAlign + Itermax	0.775	0.837	0.805	0.203	0.775	0.676	0.722	0.278	0.526	0.455	0.488	0.512	0.669	0.807	0.732	0.269	0.481	**0.788**	0.597	0.406	0.645	0.713	0.669	0.334

Table 9: Word alignment development results (P, R, F1, AER). Fixed bandwidths: MDPAlign strict k=50, MDPAlign fuzzy k=150, CTFAlign w=8. Sentence-level rows are upper-bound baselines; best per column underlined. Document-level best per column **bold**.

Model	en-fr				en-ro				en-ja				en-zh				la-gr				en-cz				Avg			
	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER	P	R	F1	AER
SENTENCE-LEVEL																												
LaBSE SimAlign + Argmax	0.969	0.948	0.958	0.040	0.921	0.676	0.780	0.220	0.755	0.456	0.569	0.431	0.880	0.819	0.848	0.151	0.710	0.707	0.708	0.292	0.883	0.899	0.891	0.109	0.853	0.751	0.792	0.207
mmBERT SimAlign + Argmax	0.958	0.932	0.945	0.053	0.906	0.639	0.749	0.251	0.738	0.419	0.535	0.465	0.867	0.757	0.808	0.192	0.690	0.711	0.700	0.300	0.878	0.856	0.867	0.133	0.840	0.719	0.767	0.232
Qwen SimAlign + Argmax	0.932	0.826	0.876	0.117	0.795	0.527	0.634	0.366	0.647	0.354	0.458	0.542	0.843	0.670	0.747	0.253	0.646	0.774	0.704	0.297	0.809	0.796	0.802	0.197	0.779	0.658	0.704	0.295
GPT-5.4-mini 1-shot prompt	0.831	0.787	0.808	0.185	0.552	0.574	0.563	0.437	0.270	0.599	0.372	0.628	0.543	0.674	0.601	0.399	0.788	0.861	0.823	0.179	0.638	0.678	0.657	0.344	0.604	0.696	0.637	0.362
DOCUMENT-LEVEL																												
GPT-5.4-mini																												
1-shot prompt	0.006	0.005	0.005	0.995	0.032	0.031	0.031	0.968	0.011	0.019	0.014	0.986	0.057	0.076	0.065	0.935	0.004	0.005	0.004	0.996	0.019	0.018	0.018	0.982	0.022	0.026	0.023	0.977
LaBSE																												
SimAlign + Argmax	0.940	0.859	0.898	0.097	0.905	0.596	0.719	0.281	0.630	0.253	0.361	0.639	0.842	0.715	0.773	0.227	0.446	0.308	0.364	0.635	0.873	0.791	0.830	0.168	0.773	0.587	0.657	0.341
MDPAlign + Argmax (fuzzy)	0.938	0.847	0.890	0.103	**0.926**	0.658	0.769	0.231	0.567	0.275	0.370	0.630	0.857	0.800	**0.828**	**0.172**	**0.653**	0.598	0.624	0.375	0.877	0.868	0.872	0.127	0.803	0.674	0.725	0.273
MDPAlign + Itermax (fuzzy)	0.887	0.894	0.890	0.110	0.809	**0.748**	**0.777**	**0.222**	0.477	0.350	0.404	0.596	0.747	**0.882**	0.809	0.191	0.533	0.680	0.598	0.405	0.777	**0.931**	0.847	0.158	0.705	0.748	0.721	0.280
CTFAlign + Argmax	**0.958**	0.899	**0.928**	**0.068**	0.921	0.653	0.764	0.235	**0.706**	0.378	0.492	0.507	0.854	0.791	0.821	0.179	0.651	0.601	0.625	0.375	**0.879**	0.869	**0.874**	**0.126**	**0.828**	0.699	**0.751**	**0.248**
CTFAlign + Itermax	0.904	**0.934**	0.919	0.084	0.807	0.743	0.774	0.226	0.597	**0.459**	**0.519**	**0.481**	0.743	0.869	0.801	0.199	0.522	0.658	0.582	0.419	0.779	0.927	0.847	0.159	0.725	**0.765**	0.740	0.261
mmBERT																												
SimAlign + Argmax	0.742	0.418	0.535	0.457	0.866	0.419	0.565	0.435	0.566	0.115	0.191	0.809	0.854	0.598	0.703	0.296	0.274	0.125	0.172	0.828	0.811	0.474	0.598	0.400	0.686	0.358	0.461	0.537
MDPAlign + Argmax (fuzzy)	0.803	0.684	0.739	0.254	**0.890**	0.563	0.690	0.310	0.404	0.166	0.235	0.764	**0.863**	0.745	**0.800**	**0.200**	0.458	0.448	0.453	0.547	0.819	0.698	0.754	0.245	0.706	0.551	0.612	0.387
MDPAlign + Itermax (fuzzy)	0.670	0.779	0.720	0.288	0.771	**0.646**	**0.703**	**0.297**	0.328	0.214	0.259	0.741	0.736	**0.849**	0.788	0.212	0.372	0.553	0.445	0.558	0.693	0.804	0.744	0.258	0.595	0.641	0.610	0.392
CTFAlign + Argmax	0.758	0.576	0.655	0.332	0.878	0.549	0.676	0.325	0.542	0.216	0.309	0.691	0.847	0.711	0.773	0.227	0.226	0.173	0.196	0.803	**0.830**	0.693	0.755	0.243	0.680	0.486	0.561	0.437
CTFAlign + Itermax	0.513	0.451	0.480	0.515	0.754	0.605	0.671	0.329	0.462	0.292	0.358	0.642	0.729	0.810	0.767	0.233	0.275	0.359	0.311	0.690	0.707	0.796	0.749	0.252	0.573	0.552	0.556	0.444
Qwen																												
SimAlign + Argmax	0.856	0.644	0.735	0.252	0.837	0.510	0.634	0.366	**0.631**	0.291	0.398	0.602	0.802	0.649	0.717	0.282	0.564	0.550	0.557	0.443	0.828	0.757	0.791	0.208	0.753	0.567	0.639	0.359
MDPAlign + Argmax (fuzzy)	0.837	0.691	0.757	0.232	0.832	0.545	0.659	0.342	0.401	0.180	0.248	0.752	0.788	0.708	0.746	0.254	0.556	0.662	0.604	0.397	0.806	0.752	0.778	0.221	0.703	0.590	0.632	0.366
MDPAlign + Itermax (fuzzy)	0.750	0.777	0.763	0.239	0.715	0.631	0.670	0.330	0.330	0.231	0.272	0.728	0.662	0.811	0.729	0.271	0.436	0.739	0.548	0.455	0.672	0.839	0.746	0.257	0.594	0.671	0.621	0.380
CTFAlign + Argmax	**0.895**	0.792	**0.840**	**0.151**	0.833	0.548	0.661	0.339	0.629	0.369	0.465	0.535	0.789	0.706	0.745	0.255	**0.643**	0.790	**0.709**	**0.293**	0.810	0.809	**0.809**	**0.191**	**0.767**	0.669	**0.705**	**0.294**
CTFAlign + Itermax	0.792	**0.863**	0.826	0.181	0.711	0.631	0.669	0.332	0.519	**0.449**	**0.481**	**0.518**	0.660	0.802	0.724	0.276	0.505	**0.845**	0.632	0.371	0.685	**0.881**	0.771	0.233	0.645	**0.745**	0.684	0.319

Table 10: Detailed word alignment test set results (P, R, F1, AER) across language pairs and models. Sentence-level rows are upper-bound baselines; best per column underlined. Document-level best per column **bold**.