BiMax: Bidirectional MaxSim Score for Document-Level Alignment

Xiaotian Wang[1,2] **Takehito Utsuro**[1] **Masaaki Nagata**[3]
[1]University of Tsukuba [2]University of Tokyo
[3]NTT Communication Science Laboratories, NTT Corporation, Japan
[1,2]wangxiaotian1999@_outlook.com, [1]utsuro@_iit.tsukuba.ac.jp
[3]masaaki.nagata@_ntt.com

Abstract

Document alignment is necessary for the hierarchical mining (Bañón et al., 2020; Morishita et al., 2022), which aligns documents across source and target languages within the same web domain. Several high-precision sentence embedding-based methods have been developed, such as TK-PERT (Thompson and Koehn, 2020) and Optimal Transport (OT) (Clark et al., 2019; El-Kishky and Guzmán, 2020). However, given the massive scale of web mining data, both accuracy and speed must be considered. In this paper, we propose a cross-lingual **Bi**directional **Max**sim score (BiMax) for computing doc-to-doc similarity, to improve efficiency compared to the OT method. Consequently, on the WMT16 bilingual document alignment task, BiMax attains accuracy comparable to OT with an approximate 100-fold speed increase. Meanwhile, we also conduct a comprehensive analysis to investigate the performance of current state-of-the-art multilingual sentence embedding models. All the alignment methods in this paper are publicly available as a tool called *EmbDA*[1].

1 Introduction

Document alignment is the task of finding parallel document pairs, which are identified as translations of each other, within a collection of documents. It is mainly employed as a preparatory stage within hierarchical mining for parallel sentence pair curation (Bañón et al., 2020; Morishita et al., 2022; Nagata et al., 2024), seeking to enhance pair quality (Sloto et al., 2023; Steingrimsson, 2023) by restricting sentence alignment in high-precision aligned document pairs. With recent advances in document-level machine translation (Sun et al., 2022; Wang et al., 2023, 2024b; Pal et al., 2024), document alignment has also become a viable strategy for developing high-quality parallel document pairs (Suryanarayanan et al., 2024).

There are four mainstream approaches: URL matching (Germann, 2016; Papavassiliou et al., 2016), bilingual lexicon (Azpeitia and Etchegoyhen, 2016; Medved' et al., 2016), machine translation (Dara and Lin, 2016; Buck and Koehn, 2016b), sentence embedding (Clark et al., 2019; Thompson and Koehn, 2020; El-Kishky and Guzmán, 2020).

Wang et al. (2024c) proposed the Overlapping Fixed-Length Segmentation (OFLS) as an alternative to Sentence-based Segmentation (SBS) for generating embeddings. When applied to Mean-Pool, TK-PERT (Thompson and Koehn, 2020), and OT (Clark et al., 2019; El-Kishky and Guzmán, 2020), this strategy led to both speed and accuracy improvements. Among these methods, OT achieves the highest recall in the WMT16 bilingual document alignment shared task (Buck and Koehn, 2016a) based on LaBSE (Feng et al., 2022). However, the computation of OT inherently involves an optimization process, necessitating multiple iterative operations. This results in high computational complexity, limiting its performance in speed.

Thus, we propose the **Bi**directional **Max**Sim score (BiMax), which matches the maximum similarity between a given segment and the opposed segment collection and then sums and averages the similarity scores. The implementation is computationally efficient, requiring only a single similarity matrix computation followed by two max-pooling operations. This idea is inspired by the MaxSim Score in ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022), which uses a late interaction mechanism to reduce the query-passage computational cost by calculating only the maximum similarity for each query token relative to the tokens in the passage. We extend this score to the sentence level and make it bidirectional.

Additionally, we evaluate combinations of state-of-the-art embedding models (i.e., models that perform well in tasks such as bitext mining and STS) with various segmentation strategies and document

[1]https://github.com/EternalEdenn/EmbDA

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13095–13116
November 4-9, 2025

alignment methods on the small-scale Ja-En MnRN dataset (Wang et al., 2024c), aiming to find suitable models and methods for different scenarios. Meanwhile, we make a modest attempt to examine the performance of different methods on low-resource languages. Finally, we build a downstream MT benchmark[2] to assess the impact of document alignment based on the WMT23 Parallel Data Curation task, and the construction process is comprehensively and transparently recorded in Appendix G.

2 Related Work

Currently, there are four mainstream approaches to document alignment. The first involves simply calculating similarity based on the URLs of the documents (Germann, 2016; Papavassiliou et al., 2016). The second uses a bag-of-words or bag-of-ngrams representation of the document contents, leveraging a bilingual lexicon for computation (Azpeitia and Etchegoyhen, 2016; Medved' et al., 2016). The third entails translating documents into the same language, followed by similarity calculations using ngram-based metrics (e.g., BLEU) (Dara and Lin, 2016; Buck and Koehn, 2016b). The fourth utilizes multilingual pre-trained embedding models to map documents into a shared vector space, where similarity is determined by the distances between vectors (Clark et al., 2019; Thompson and Koehn, 2020; El-Kishky and Guzmán, 2020). In the WMT16 bilingual document alignment shared task (Buck and Koehn, 2016a), numerous techniques and systems were proposed. However, due to the limitations of technology at the time, all efforts focused on the first three approaches, with no exploration of embedding-based methods.

With the development of pre-trained multilingual sentence embedding models (Artetxe and Schwenk, 2019; Feng et al., 2022), which map sentences from different languages into a shared multilingual vector space, cross-lingual bitext mining has become feasible. This progress also facilitates representing documents using segment embeddings and computing doc-pair similarities via vector-based methods.

Thompson and Koehn (2020) introduced TK-PERT, a method that assigns weights to sentences using regionally emphasized windows derived from a modified PERT distribution (Vose, 2000) to form document feature vectors. Optimal Transport (OT) was also applied in cross-lingual document alignment, evolving from the word level with Word Movers' Distance (WMD) (Kusner et al., 2015) to the sentence level with Sentence Movers' Distance SMD) and Greedy Movers' Distance (GMD) (Clark et al., 2019; El-Kishky and Guzmán, 2020). Building on GMD, Fernando et al. (2023) et al. employed a new weighting strategy using bilingual lexicons, further enhancing alignment accuracy in low-resource languages. Wang et al. (2024c) proposed OFLS instead of SBS for the embedding step. However, their work is limited to using only the LaBSE model and does not explore new document alignment methods.

3 Method

Unlike MaxSim utilized in ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022), which uses the query's hidden word embeddings to search for the most similar token in the passage undirectionally, we apply it to sentence-level as the Bidirectional MaxSim Score (BiMax), introducing the following key modifications: (1) transforming from monolingual to cross-lingual, (2) shifting from word-level embeddings to sentence-level embeddings, and (3) moving from one-sided maximum similarity matching to a bidirectional approach.

3.1 Bidirectional MaxSim Score

We define the source / target document set as $\mathcal{D}_S$ and $\mathcal{D}_T$, and adopt a 2-stage approach to consider the $\mathcal{D}_S \times \mathcal{D}_T$ possible document pairs:

1. **Candidate Generation**: We first use MeanPool or TK-PERT method to generate a single feature vector for each document, and then employ Faiss Search (Johnson et al., 2019) to retrieve K target documents as potential matches for each source document.
2. **Candidate Re-ranking**: We re-rank the $\mathcal{D}_S \times K$ pairs using a more accurate but slower and sometimes more memory-intensive scoring method, such as OT and our proposed BiMax.

Let s_i for $i \in \{0, ..., N_S - 1\}$ be the N_S segments in a given source document S and t_j for $j \in \{0, ..., N_T - 1\}$ be the N_T segments in a given target document T. The BiMax Score is defined as:

$$\text{MaxSim}(S, T) = \frac{1}{N_S} \sum_{i=1}^{N_S} \max_{t \in T} \text{Sim}(s_i, t) \tag{1a}$$

$$\text{BiMax}(S, T) = \frac{1}{2}(\text{MaxSim}(S, T) + \text{MaxSim}(T, S)) \tag{1b}$$

where $\text{Sim}(s, j)$ represents the similarity score. In this work, we use a pre-trained multilingual sentence embedding model to map the source segment

[2]This benchmark is offered solely as a reference rather than a definitive proposal, thus we include it only in Appendix G.

Strategies & Models		LaBSE	distiluse-base-multi-cased-v2	BGE M3	jina-embed-v3
Common Info. (Source / Target)					
Document Num.		232 / 931			
Total Sentence Num.		4,746 / 57,032			
Gold Pairs		263[3]			
Total Document Token Num.		0.50M / 3.34M	0.53M / 3.68M	0.43M / 3.68M	0.43M / 3.68M
Average Sentence Token Num.		105.17 / 58.55	111.27 / 64.49	90.78 / 64.48	90.78 / 64.48
Distinct Info. (Source / Target)					
SBS	Segment Num.	4,746 / 57,032			
	Avg Seg Len.	105.17 / 58.55	111.27 / 64.49	90.78 / 64.48	90.78 / 64.48
	MP PPROC: Time	131.33s	80.42s	640.22s	133.01s
	MP PPROC: Memory	4455.53 MB.	7267.58 MB.	57924.36 MB.	7036.57 MB.
	TK PPROC: Time	206.19s	164.89s	745.57s	247.22s
	TK PPROC: Memory	4478.97 MB.	7291.22 MB.	57948.21 MB.	7052.71 MB.
Blob (Max 64)	Segment Num.	4,083 / 38,828	4,189 / 40,971	3,752 / 41,706	3,752 / 41,706
	Avg Seg Len.	122.24 / 86.01	126.06 / 89.76	114.83 / 88,17	114.83 / 88,17
	MP PPROC: Time	107.54s	70.92s	564.88s	127.97s
	MP PPROC: Memory	4392.51 MB.	7213.87 MB.	55890.82 MB.	7023.15 MB.
	TK PPROC: Time	164.87s	139.41s	655.54s	220.72s
	TK PPROC: Memory	4416.13 MB.	7238.25 MB.	55914.12 MB.	7040.32 MB.
OFLS (FL 30, OR 0.5)	Segment Num.	33,151 / 222,149	35,082 / 244,688	28,594 / 244,653	28,594 / 244,653
	Avg Seg Len.	29.95 / 29.97	29.95 / 29.97	29.95 / 29.97	29.95 / 29.97
	MP PPROC: Time	71.38s	49.25s	119.36s	380.51s
	MP PPROC: Memory	2758.95 MB.	1685.84 MB.	2338.35 MB.	3203.90 MB.
	TK PPROC: Time	569.54s	591.48s	650.14s	912.74s
	TK PPROC: Memory	2782.64 MB.	1715.25 MB.	2370.38 MB.	3236.67 MB.

Table 1: The statistical information regarding the preprocessing steps before document alignment, where "PPROC" represents for preprocessing.

s and the target segment t into the same vector space, producing embeddings E_s and E_t, and then adopting their cosine similarity $cos(E_s, E_t)$.

4 Analysis of Document Alignment Performance on MnRN

We use the Ja-En MnRN dataset to conduct the analysis under various sentence embedding models, three segmentation strategies, SBS[4], Blob[5] (Finkelstein et al., 2024), and OFLS[6] (Wang et al., 2024c), and four document alignment methods, focusing on three main points: (1) which model is suitable for which segmentation strategy, (2) how do different document alignment methods perform under each model, and (3) which combination of these three factors yields the best results.

The reasons for selecting embedding models and the model settings are recorded in Appendix A and B, while the experimental setup and the details of the evaluation metrics are described in Appendix C.

We present the statistical information for four models under various segmentation strategies in Table 1. Since Mean-Pool, OT w/Mean, and BiMax w/Mean require the same preprocessing steps for document alignment, which include segmentation, segment embedding, and mean vector generation, we only use Mean-Pool (MP) as a representative. In contrast, TK-PERT (TK) incurs additional time for LIDF[7] and the modified PERT distribution compared to MP, resulting in a longer preprocessing time that is dependent on the number of segments.

4.1 Performance Comparison

(1) *Which model is suitable for which segmentation strategy?*

As shown in Table 2, we present the results of five models labeled (a)∼(e). More detailed results for additional models can be found in Table 6 of Appendix A. For models (a), (b), and (d), OFLS demonstrates an improvement in the F1 score in most cases and a reduction in preprocessing time (except for TK-PERT) compared to the other two segmentation strategies. However, for the LASER-2 model, although the use of OFLS improves the accuracy of the TK-PERT and Bi-

[3]Because the English documents contain duplicates, the number of gold pairs exceeds that of the Japanese documents.

[4]Sentence-based Segmentation (SBS): split a document into sentences using delimiters such as line breaks or periods.

[5]Blob: concatenate multiple consecutive sentences as a single unit until reaching a specified limitation.

[6]Overlapping Fixed-Length Segmentation (OFLS): split a document into segments through a fixed-length sliding window, with a proportion of overlap between adjacent segments.

[7]LIDF is used for scaling segments based on the inverse of the (linear, rather than logarithmic) number of documents that contain the given segment.

Strategies & Models		Embedding Models				
		(a) LaBSE	(b) distiluse-base-multi-cased-v2	(c) LASER-2	(d) BGE M3 (dense only)	(e) jina-embeddings-v3
Experiments (F1 Score ↑ / PPROC. Time (sec.) ↓)						
SBS	Mean-Pool	0.8362 / 131.27s	0.8362 / 80.40s	0.5862 / 543.10s	0.8448 / 637.01s	0.8362 / 133.72s
	TK-PERT	0.8448 / 206.19s	0.8147 / 164.89s	0.5819 / **652.32s**	0.8362 / 745.57s	0.8706 / 247.22s
	OT w/Mean	0.8448 / 131.58s	0.8448 / 80.46s	**0.4784** / 543.87s	0.8621 / 642.20s	0.8578 / 132.73s
	BiMax w/Mean	$0.8922^{\dagger\ddagger}$ / 131.47s	$0.9052^{\dagger\ddagger}$ / 80.49s	$0.7414^{\dagger\ddagger}$ / 543.61s	$0.9181^{\dagger\ddagger}$ / 640.27s	**$0.9310^{\dagger\ddagger}$** / 134.52s
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s	**0.8663** / 70.80s	**0.5948** / **533.63s**	**0.8750** / 565.45s	**0.8448** / **127.75s**
	TK-PERT	0.8663 / **164.87s**	0.8491 / **139.41s**	0.5905 / **640.51s**	0.8534 / 655.54s	0.8578 / **220.72s**
	OT w/Mean	0.8233 / 107.84s	0.8405 / 70.46s	0.4439 / **533.61s**	0.8362 / 564.84s	0.8276 / **128.12s**
	BiMax w/Mean	$0.9009^{\dagger\ddagger}$ / 106.65s	$0.9052^{\dagger\ddagger}$ / 71.16s	$0.7586^{\dagger\ddagger}$ / **533.08s**	$0.9181^{\dagger\ddagger}$ / 564.76s	$0.9052^{\dagger\ddagger}$ / **127.32s**
OFLS (FL 30, OR 0.5)	Mean-Pool	**0.8707** / **71.59s**	0.8233 / **49.23s**	0.5302 / 1246.64s	0.8491 / **119.38s**	0.7716 / 380.98s
	TK-PERT	**0.9483** / 569.54s	**0.8966** / 591.48s	**0.8134** / 1860.80s	**0.9224** / **650.14s**	**0.9310** / 912.74s
	OT w/Mean	**0.9569** / **71.33s**	**0.9397** / **49.10s**	0.4354 / 1223.61s	**0.8879** / **119.36s**	**0.8966** / 379.59s
	BiMax w/Mean	**0.9612** / **71.14s**	**$0.9569^{\dagger}$** / **49.32s**	**$0.7845^{\ddagger}$** / 1225.91s	**$0.9483^{\ddagger}$** / **119.36s**	$0.9267^{\ddagger}$ / 381.05s

Table 2: The results for comparing SBS, Blob, and OFLS under each embedding model on the Ja-En MnRN dataset, where "FL" represents for fixed-length, "OR" represents for overlapping rate, "Max" represents the token limitation of Blob. For each model and the four document alignment methods, we underline and bold the **result** that achieves the higher F1 score or shorter preprocessing time under SBS, Blob, or OFLS. For each segmentation strategy within each model, † is appended when BiMax demonstrates statistically significant superiority over both Mean-Pool and TK-PERT, and ‡ is used when it is significantly superior to OT.

Max methods, its performance on Mean-Pool and OT remains poor. Additionally, the preprocessing speed is obviously diminished, which may be attributed to the chain structure of LSTM, due to the rise in the total number of tokens resulting from overlapping segments in OFLS.

Specifically, the jina-embeddings-v3 model achieves a relatively high F1 score when using the SBS segmentation, with a comparable speed to LaBSE. Although employing OFLS may further enhance accuracy, the preprocessing time for the jina-embeddings-v3 model becomes longer, which may be caused by the use of RoPE (Su et al., 2024) and FlashAttention 2 (Dao, 2024) mechanisms.

Moreover, we provide an expanded discussion of Blob in Appendix D.

(2) *How do different document alignment methods perform under each model?*

Due to the limited scale of the MnRN dataset, the similarity computation times across different segmentation strategies and embedding models show minimal variation across the four document alignment methods. Therefore, we present the distribution of these times in Figure 1, while Appendix A provides detailed results.

Figure 1 shows that the similarity computation time required for BiMax is much shorter than OT. However, it should be noted that, OT processes document pairs sequentially due to its optimization routine. In contrast, BiMax supports batched parallel computation. For a fair runtime comparison, BiMax is limited to single-pair computation in this paper. We follow Yeh (2000) and conduct a statistical significance test ($p < 0.05$) between BiMax and the other three document alignment methods. The detailed process is described in Appendix C. Despite its lightweight design, BiMax outperforms competing methods in most scenarios and shows statistically significant gains in some cases.

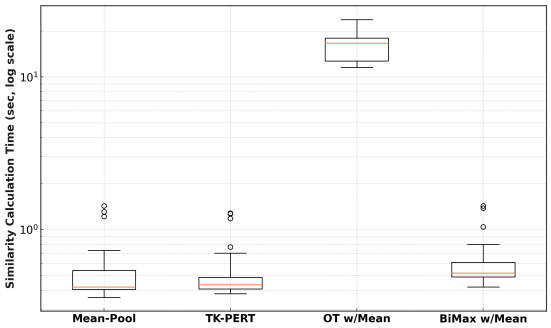

Figure 1: The similarity computation time (in seconds, log scale) for the four document alignment methods.

(3) *Which combination of these three factors yields the best results?*

Overall, when using OFLS, LaBSE demonstrates superior accuracy compared to other models, and among the document alignment methods, according to Table 2, BiMax achieves the best performance. The model closest to LaBSE under OFLS, distiluse-base-multilingual-cased-v2, while lower in accuracy, offers advantages in terms of speed and memory efficiency according to Table 1.

5 Experiment on the WMT16 document alignment shared task

To test the BiMax method further, we conduct experiments on the WMT16 document alignment task. We use the same settings as Wang et al. (2024c) for

a comparison with their work. The detailed experimental setup and dataset information are recorded in Appendix C.

The results are presented in Table 3. Similarly, under the OFLS segmentation, the BiMax method improves recall by 0.3% to 2.4% compared to SBS. Compared with the results of Wang et al. (2024c), the BiMax method demonstrates slightly higher accuracy than the OT and TK-PERT methods under SBS. However, the opposite trend is observed under OFLS. Although BiMax cannot comprehensively outperform OT, its speed achieves approximately a 100-fold increase relative to OT, measured by the number of document pairs processed per second. Rather than solely prioritizing precision, this research emphasizes the efficiency of the method. Moreover, as noted in Section 4.1, BiMax can be executed in parallel via batch processing, potentially resulting in faster similarity computations.

Method	Segment Strategy	Recall↑	Sim Speed (pairs / sec.)↑
Wang et al. (2024c) (LaBSE)			
Mean-Pool	SBS	82.6%	97,358.98
Mean-Pool	OFLS	92.6%	
TK-PERT	SBS	95.2%	
TK-PERT	OFLS	96.3%	
OT w/Mean-Pool	SBS	90.6%	91.98
OT w/TK-PERT	SBS	95.6%	
OT w/Mean-Pool	OFLS	**93.7%**	99.34
OT w/TK-PERT	OFLS	**96.8%**	
This work (LaBSE)			
BiMax w/Mean-Pool	SBS	**90.7%**	**11,510.92**
BiMax w/TK-PERT	SBS	**95.8%**	
BiMax w/Mean-Pool	OFLS	93.1%	**13,220.15**
BiMax w/TK-PERT	OFLS	96.1%	

Table 3: The results of soft recall on the WMT16 test data. Between BiMax and OT, we highlight the superior result in **bold**.

6 Experiments on Low-Resource Languages

We use the dataset constructed by Fernando et al. (2023), which covers English, Sinhala, and Tamil (hereafter referred to as the Fernando dataset[8]) to evaluate the effectiveness of BiMax. The dataset comprises four web domains: Army, Hiru, ITN, and NewsFirst, and three language pairs: En–Si, En–Ta, and Si–Ta. More detailed experimental settings and dataset statistics are provided in Appendix C and Appendix E.

We conduct experiments on three language pairs across four web domains. For each language pair, the final recall is computed as the weighted average of its results over the domains, with weights determined by the number of gold pairs in each domain. Detailed results are documented in Appendix E.

Method	Segment Strategy	Language Pair		
		En-Si	En-Ta	Si-Ta
LaBSE				
Mean-Pool	SBS	93.10%	77.70%	79.41%
	OFLS	92.97%	86.39%	85.13%
TK-PERT	SBS	89.32%	74.52%	75.48%
	OFLS	90.18%	82.64%	80.98%
OT	SBS	91.83%	78.94%	81.24%
	OFLS	92.51%	86.12%	86.74%
BiMax	SBS	**95.53%**	83.85%	84.91%
	OFLS	95.41%	**91.33%**	**89.71%**

Table 4: The results of recall on the Fernando dataset. We highlight the best one in **bold** for each language pair.

As shown in Table 4, BiMax outperforms the other three methods across all three language pairs, with a pronounced improvement observed in En–Ta. Furthermore, in most cases, OFLS achieves better performance than SBS. Even in the case of En–Si, where OFLS is slightly inferior to SBS, the difference remains marginal.

Meanwhile, as a multi-way dataset, the same method exhibits considerable performance variation across different language pairs. In particular, when Tamil is used as the target language for retrieval, the accuracy differs substantially compared with En–Si, a discrepancy that may be attributed to variations in the embedding precision of the model across different languages. Moreover, considering the dataset-specific characteristics, factors beyond language, such as document length, may also affect alignment accuracy. Specifically, TK-PERT and OT are possibly better suited for handling long texts but perform less effectively on short texts. A more detailed analysis is provided in Appendix F.

7 Conclusion

This paper introduces a novel and efficient BiMax Score for the document alignment task, reducing computational complexity compared to OT. However, while BiMax shows the best performance on the Fernando dataset and the small-scale MnRN dataset, results from the WMT16 document alignment task reveal that we cannot definitively assert BiMax's accuracy surpasses OT or TK-PERT. Instead, we advocate for BiMax primarily for its efficiency in scenarios such as processing large-scale web-crawled data. In these cases, according to our analysis of experiments, the LaBSE + OFLS + BiMax approach is recommended, as it outperforms all other combinations.

[8] https://github.com/kdissa/comparable-corpus

8 Limitations

The existing publicly available datasets for document alignment are limited. Even large-scale multilingual parallel document corpora such as CC-Aligned[9] (El-Kishky et al., 2020), which consist of web pages aligned through automated document alignment methods, cannot guaranty ground truth due to the absence of manual verification. In addition, although we have explored the effectiveness of BiMax on low-resource languages, the Fernando dataset (Fernando et al., 2023) covers only Sinhala and Tamil. Since low-resource languages differ considerably from one another, it cannot be guaranteed that the method generalizes equally well to all such languages. Moreover, many other low-resource languages still lack established datasets.

Furthermore, although we evaluated multiple embedding models on the Ja-En MnRN dataset, the representational capabilities of different embedding models vary across languages. Therefore, the LaBSE model may not consistently achieve optimal performance in all scenarios.

Finally, as discussed in Section 5, Section 6 and Section 7, its performance under OFLS does not surpass TK-PERT and OT on the WMT16 document alignment task. Thus, we emphasize efficiency rather than solely pursuing precision.

9 Ethical statement

The embedding models used in this paper, LaBSE (Feng et al., 2022), LASER-2 (Heffernan et al., 2022), LEALLA (Mao and Nakagawa, 2023), paraphrase-multilingual-MiniLM-L12-v2, and distiluse-base-multilingual-cased-v2, paraphrase-multilingual-mpnet-base-v2 (Reimers and Gurevych, 2019), BGE M3 (Chen et al., 2024), and jina-embeddings-v3 (Sturua et al., 2024), are publicly available for research.

The WMT16 test data is provided by the WMT16 document alignment shared task (Buck and Koehn, 2016a), and the Fernando dataset has been publicly released by Fernando et al. (2023).

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Andoni Azpeitia and Thierry Etchegoyhen. 2016. DOCAL - vicomtech's participation in the WMT16 shared task on bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 666–671, Berlin, Germany. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Christian Buck and Philipp Koehn. 2016a. Findings of the WMT 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.

Christian Buck and Philipp Koehn. 2016b. Quick and reliable document alignment via TF/IDF-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678, Berlin, Germany. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Aswarth Abhilash Dara and Yiu-Chang Lin. 2016. YODA system for WMT16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 679–684, Berlin, Germany. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document

[9] https://www.statmt.org/cc-aligned/

pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Ahmed El-Kishky and Francisco Guzmán. 2020. Massively multilingual document alignment with cross-lingual sentence-mover's distance. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 616–625, Suzhou, China. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Aloka Fernando, Surangika Ranathunga, Dilan Sachintha, Lakmali Piyarathna, and Charith Rajitha. 2023. Exploiting bilingual lexicons to improve multilingual embedding-based document and sentence alignment for low-resource languages. *Knowledge and Information Systems*, 65(2):571.

Mara Finkelstein, David Vilar, and Markus Freitag. 2024. Introducing the NewsPaLM MBR and QE dataset: LLM-generated high-quality parallel data outperforms traditional web-crawled data. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1355–1372, Miami, Florida, USA. Association for Computational Linguistics.

Ulrich Germann. 2016. Bilingual document alignment with latent semantic indexing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 692–696, Berlin, Germany. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

J. Johnson, M. Douze, and H. Jégou. 2019. Billion-scale similarity search with GPUs. *Journal 2019 IEEE*, pages 535–547.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, China. ACM.

M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. 2015. From word embeddings to document distances. In *Proc 32nd PRML*, pages 957–966.

Zhuoyuan Mao and Tetsuji Nakagawa. 2023. LEALLA: Learning lightweight language-agnostic sentence embeddings with knowledge distillation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1886–1894, Dubrovnik, Croatia. Association for Computational Linguistics.

Marek Medved', Miloš Jakubíček, and Vojtech Kovář. 2016. English-French document alignment based on keywords and statistical translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 728–732, Berlin, Germany. Association for Computational Linguistics.

Nguyen-Hoang Minh-Cong, Nguyen Van Vinh, and Nguyen Le-Minh. 2023. A fast method to filter noisy parallel data WMT2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 359–365, Singapore. Association for Computational Linguistics.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France. European Language Resources Association.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. 2024. JaParaPat: A large-scale Japanese-English parallel patent application corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9452–9462, Torino, Italia. ELRA and ICCL.

Proyag Pal, Alexandra Birch, and Kenneth Heafield. 2024. Document-level machine translation with large-scale public parallel corpora. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13185–13197, Bangkok, Thailand. Association for Computational Linguistics.

Vassilis Papavassiliou, Prokopis Prokopidis, and Stelios Piperidis. 2016. The ILSP/ARC submission to the WMT 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 733–739, Berlin, Germany. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sonal Sannigrahi, Josef van Genabith, and Cristina España-Bonet. 2023. Are the best multilingual document embeddings simply based on sentence embeddings? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2306–2316, Dubrovnik, Croatia. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Steve Sloto, Brian Thompson, Huda Khayrallah, Tobias Domhan, Thamme Gowda, and Philipp Koehn. 2023. Findings of the WMT 2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 95–102, Singapore. Association for Computational Linguistics.

S. Steingrimsson. 2023. A sentence alignment approach to document alignment and multi-faceted filtering for curating parallel sentence pairs from web-crawled data. In *Proc 8th WMT*, pages 366–374.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548, Dublin, Ireland. Association for Computational Linguistics.

Sanjay Suryanarayanan, Haiyue Song, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M Khapra, and Raj Dabre. 2024. Pralekha: An indic document alignment evaluation benchmark. *arXiv preprint arXiv:2411.19096*.

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, Hong Kong, China. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2020. Exploiting sentence order in document alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

D Vose. 2000. Risk analysis: a quantitative guide. John Wiley & Sons.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Longyue Wang, Zefeng Du, Wenxiang Jiao, Chenyang Lyu, Jianhui Pang, Leyang Cui, Kaiqiang Song, Derek Wong, Shuming Shi, and Zhaopeng Tu. 2024b. Benchmarking and improving long-text translation with large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7175–7187, Bangkok, Thailand. Association for Computational Linguistics.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023.

Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Xiaotian Wang, Takehito Utsuro, and Masaaki Nagata. 2024c. Document alignment based on overlapping fixed-length segments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 51–61, Bangkok, Thailand. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

A Embedding Model Selection

In Section 4, first, we choose the LaBSE (Feng et al., 2022) and LASER-2 models (Heffernan et al., 2022), which are frequently used for the bitext mining task, and also include a knowledge-distilled, lightweight variant of LaBSE, the LEALLA model (Mao and Nakagawa, 2023).

Subsequently, we employ three representative multilingual models from the Sentence Transformers library[10]: paraphrase-multilingual-MiniLM-L12-v2, distiluse-base-multilingual-cased-v2, and paraphrase-multilingual-mpnet-base-v2 (Reimers and Gurevych, 2019), which perform strongly on the STS task.

Finally, considering the MTEB benchmark (Muennighoff et al., 2023), which encompasses several embedding tasks, we select two models that currently achieve state-of-the-art performance on the leaderboard[11], which are capable of processing long sentences and suitable for multi-task scenarios: BGE M3 (Chen et al., 2024), and jina-embeddings-v3 (Sturua et al., 2024). Additionally, we also consider the multi-e5-large model (Wang et al., 2024a) and the paraphrase-multilingual-mpnet-base-v2 model (Reimers and Gurevych, 2019). The results are presented in Table 6.

B Embedding Model Settings

We maintain the default configurations for all models, as these configurations represent the most general use cases. However, BGE M3 employs a half-precision floating-point format (fp16) by default, whereas most other models utilize a single-precision floating-point format (fp32). Furthermore, BGE M3 and LASER-2 generate vectors in the form of NumPy arrays, while other models predominantly output tensors or offer tensor output as an option. To establish method consistency, we implement a standardization protocol, converting all vectors to fp32 format and utilizing tensors after the embedding process.

Meanwhile, given that all models except LASER-2 are derived from Hugging Face[12] , we can achieve substantial uniformity in the Python library and code framework, thereby facilitating meaningful comparisons of inference speeds across models. However, due to the LASER-2 model's different library and code program, absolute parity in comparative speed analysis between LASER-2 and other models cannot be established.

Because of the multifunctionality of the three multi-task models, we specify distinct usage. For the multi-e5-large model, which can leverage a prefix (either "query:" or "passage:") as the start of the text, after testing with some combinations or omitting the prefix altogether, we find that appending "query:" to both the source and target produces the highest accuracy. Regarding the BGE M3 model, which provides three functions for generating different scores, we elect to use only its dense embedding as output. Finally, for the jina-embeddings-v3 model, which offers a selection among various LoRA adapters (Hu et al., 2021) depending on the desired task, we choose the "text-matching" task.

C Experiment Settings

We follow the experimental settings of Thompson and Koehn (2020) and Wang et al. (2024c), configuring the hyper-parameters for the WMT16 document alignment task and the MnRN dataset in the TK-PERT method as $J = 16, \gamma = 20$ and $J = 8, \gamma = 16$, respectively. The setting of the Fernando dataset is the same as the WMT16 test data. Here, J determines the number of windows produced by the TK-PERT method, while γ is a hyper-parameter that controls the peakedness of the modified PERT distribution. For OT, GMD, and BiMax, we retrieve 20, 32, and 32 candidates for each source document in the MnRN dataset, WMT test data, and Fernando dataset, respectively. We put the basic information of the three datasets in

[10] https://huggingface.co/sentence-transformers

[11] https://huggingface.co/spaces/mteb/leaderboard

[12] https://huggingface.co/

Strategies & Models			(a) LaBSE	(b) LEALLA-large	(c) paraphrase-multi-MiniLM-L12-v2	(d) distiluse-base-multi-cased-v2	(e) paraphrase-multi-mpnet-base-v2	(f) LASER-2	(g) multi-e5-large	(h) BGE M3	(i) jina-embed-v3
Common Info. (Source / Target)											
Document Num.							232 / 931				
Total Sentence Num.							4,746 / 57,032				
Gold Pairs							263				
Total Document Token Num.			0.50M / 3.34M	0.50M / 3.34M	0.43M / 3.68M	0.53M / 3.68M	0.43M / 3.68M	0.57M / 4.47M	0.43M / 3.68M	0.43M / 3.68M	0.43M / 3.68M
Average Sentence Token Num.			105.17 / 58.55	105.17 / 58.55	90.78 / 64.48	111.27 / 64.49	90.78 / 64.48	119.68 / 78.31	90.78 / 64.48	90.78 / 64.48	90.78 / 64.48
Distinct Info. (Source / Target)											
SBS	Segment Num.						4,746 / 57,032				
	Average Segment Len.		105.17 / 58.55	105.17 / 58.55	90.78 / 64.48	111.27 / 64.49	90.78 / 64.48	119.68 / 78.31	90.78 / 64.48	90.78 / 64.48	90.78 / 64.48
	MP PPROC:	Time	131.33s	60.78s	59.01s	80.42s	148.88s	543.20s	458.74s	640.22s	133.01s
		Memory	4455.53 MB.	1555.01 MB.	1855.53 MB.	7267.58 MB.	5894.61 MB.	3358.72 MB.	8561.57 MB.	57924.36 MB.	7036.57 MB.
	TK PPROC:	Time	206.19s	158.54s	158.38s	164.89s	223.87s	652.32s	517.99s	745.57s	247.22s
		Memory	4478.97 MB.	1562.89 MB.	1867.35 MB.	7291.22 MB.	5918.26 MB.	3406.95 MB.	8593.10 MB.	57948.21 MB.	7052.71 MB.
Blob (Max 64)	Segment Num.		4,083 / 38,828	4,083 / 38,828	3,752 / 41,706	4,189 / 40,971	3,752 / 41,706	4,198 / 46,761	3,752 / 41,706	3,752 / 41,706	3,752 / 41,706
	Average Segment Len.		122.24 / 86.01	122.24 / 86.01	114.83 / 88.17	126.06 / 89.76	114.83 / 88.17	135.30 / 95.51	114.83 / 88,17	114.83 / 88,17	114.83 / 88,17
	MP PPROC:	Time	107.54s	55.79s	59.04s	70.92s	125.11s	533.54s	371.31s	564.88s	127.97s
		Memory	4392.51 MB.	1535.52 MB.	1832.44 MB.	7213.87 MB.	5840.50 MB.	3343.34 MB.	8495.56 MB.	55890.82 MB.	7023.15 MB.
	TK PPROC:	Time	164.87s	66.17s	138.72s	139.41s	173.15s	640.54s	429.90s	655.54s	220.72s
		Memory	4416.13 MB.	1543.25 MB.	1844.26 MB.	7238.25 MB.	5864.29 MB.	3391.38 MB.	8527.09 MB.	55914.12 MB.	7040.32 MB.
OFLS (FL 30, OR 0.5)	Segment Num.		33,151 / 222,149	33,151 / 222,149	28,594 / 244,653	35,082 / 244,688	28,594 / 244,653	37,742 / 297,245	28,594 / 244,653	28,594 / 244,653	28,594 / 244,653
	Average Segment Len.		29.95 / 29.97	29.95 / 29.97	29.95 / 29.97	29.95 / 29.97	29.95 / 29.97	29.96 / 29.98	29.95 / 29.97	29.95 / 29.97	29.95 / 29.97
	MP PPROC:	Time	71.38s	52.60s	49.06s	49.25s	74.58s	1221.74s	259.11s	119.36s	380.51s
		Memory	2758.95 MB.	900.48 MB.	966.69 MB.	1685.84 MB.	2070.51 MB.	1871.28 MB.	3476.11 MB.	2338.35 MB.	3203.90 MB.
	TK PPROC:	Time	569.54s	548.93s	578.17s	591.48s	599.66s	1860.80s	745.20s	650.14s	912.74s
		Memory	2782.64 MB.	908.37 MB.	978.53 MB.	1715.25 MB.	2094.20 MB.	1920.33 MB.	3507.69 MB.	2370.38 MB.	3236.67 MB.

Table 5: The statistical information regarding the preprocessing steps before document alignment across various models and segmentation strategies, where “MP” represents for Mean-Pool, “TK” represents for TK-PERT, “PPROC” represents for preprocessing.

Info. & Methods		Embedding Models								
		(a) LaBSE	(b) LEALLA-large	(c) paraphrase-multi-MiniLM-L12-v2	(d) distiluse-base-multi-cased-v2	(e) paraphrase-multi-mpnet-base-v2	(f) LASER-2	(g) multi-e5-large	(h) BGE M3 (dense only)	(i) jina-embeddings-v3
Model Info.										
Suitable Task		Bitext.	Bitext.	STS	STS	STS	Bitext.	Multi-task	Multi-task	Multi-task
#Param.		471M	147M	118M	135M	278M	43M	560M	567M	572M
#Dim.		768	256	384	512	768	1024	1024	1024	1024
#Lang.		Multi.	Multi.	Multi.	Multi.	Multi.	Mono.	Multi.	Multi.	Multi.
#Arch.		Transformer	Transformer	Transformer	Transformer	Transformer	LSTM	Transformer	Transformer	Transformer
Experiments (F1 Score ↑ / PPROC. Time (sec.) ↓ / Sim. Time (sec.) ↓)										
SBS	Mean-Pool	0.8362 / 131.27s / 0.42s	0.3750 / 60.54s / 0.37s	0.7543 / 59.00s / 0.36s	0.8362 / 80.40s / 0.40s	0.7716 / 148.60s / 0.46s	0.5862 / 543.10s / 0.42s	0.7802 / 457.94s / 0.42s	0.8448 / 637.01s / 1.43s	0.8362 / 133.72s / 0.68s
	TK-PERT	0.8448 / 206.19s / 0.48s	**0.5129** / 158.54s / 0.39s	0.7845 / 158.38s / 0.47s	0.8147 / 164.89s / 0.42s	0.7931 / 223.87s / 0.41s	0.5819 / 652.32s / 0.41s	0.7845 / 517.99s / 0.46s	0.8362 / 745.57s / 1.27s	0.8706 / 247.22s / 0.77s
	OT w/Mean	0.8448 / 131.58s / 22.87s	0.4525 / 60.87s / 17.26s	0.7845 / 58.98s / 16.69s	0.8448 / 80.46s / 20.17s	0.7974 / 149.07s / 18.13s	0.4784 / 543.87s / 15.54s	0.8060 / 461.78s / 15.43s	0.8621 / 642.20s / 17.62s	0.8578 / 132.73s / 17.90s
	BiMax w/Mean	**0.8922**†‡ / 131.47s / 0.49s	0.4655 / 60.83s / 0.43s	**0.8319**†‡ / 59.35s / 0.42s	**0.9052**†‡ / 80.49s / 0.46s	**0.8577**†‡ / 148.40s / 0.49s	**0.7414**†‡ / 543.61s / 0.50s	**0.8750**†‡ / 462.17s / 0.80s	**0.9181**†‡ / 640.27s / 0.53s	**0.9310**†‡ / 134.52s / 0.70s
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s / 0.42s	0.3491 / 55.79s / 0.38s	0.7672 / 59.04s / 0.36s	0.8663 / 70.80s / 0.41s	0.7802 / 125.11s / 0.41s	0.5948 / 533.63s / 0.41s	0.7844 / 370.11s / 0.71s	0.8750 / 565.45s / 1.31s	0.8448 / 127.75s / 0.62s
	TK-PERT	0.8663 / 164.87s / 0.45s	**0.5129** / 119.17s / 0.38s	0.7155 / 138.72s / 0.40s	0.8491 / 139.41s / 0.39s	0.7241 / 173.15 / 0.41s	0.5905 / 640.51s / 0.42s	0.7672 / 429.90s / 0.50s	0.8534 / 655.54s / 1.28s	0.8578 / 220.72s / 0.70s
	OT w/Mean	0.8233 / 107.84s / 23.83s	0.4525 / 56.20s / 16.93s	0.7802 / 59.14s / 18.38s	0.8405 / 70.46s / 20.91s	0.7802 / 125.23s / 18.69s	0.4439 / 533.61s / 14.99s	0.7974 / 371.94s / 16.07s	0.8362 / 564.84s / 17.92s	0.8276 / 128.12s / 17.10s
	BiMax w/Mean	**0.9009**†‡ / 106.65s / 0.50s	0.4699 / 56.29s / 0.49s	**0.8147**†‡ / 59.16s / 0.43s	**0.9052**†‡ / 71.16s / 0.52s	**0.8534**†‡ / 125.21s / 0.43s	**0.7586**†‡ / 533.08s / 0.51s	**0.8707**†‡ / 371.15s / 0.64s	**0.9181**†‡ / 564.76s / 1.43s	**0.9052**†‡ / 127.32s / 0.68s
OFLS (FL 30, OR 0.5)	Mean-Pool	0.8707 / 71.59s / 0.42s	0.3836 / 52.76s / 0.44s	0.7759 / 49.06s / 0.36s	0.8233 / 49.23s / 0.43s	0.7112 / 74.56s / 0.40s	0.5302 / 1226.64s / 0.41s	0.7543 / 259.61s / 0.44s	0.8491 / 119.38s / 1.22s	0.7716 / 380.98s / 0.73s
	TK-PERT	0.9483 / 569.54s / 0.40s	**0.6034** / 548.93s / 0.40s	0.8707 / 578.17s / 0.48s	0.8966 / 591.48s / 0.42s	0.8793 / 599.66s / 0.40s	**0.8134** / 1860.80s / 0.42s	0.8534 / 745.20s / 0.45s	0.9224 / 650.14s / 1.18s	**0.9310** / 912.74s / 0.63s
	OT w/Mean	0.9569 / 71.33s / 12.67s	0.4782 / 52.47s / 12.67s	0.8578 / 49.08s / 11.64s	0.9397 / 49.10s / 12.47s	0.8922 / 74.31s / 11.54s	0.4354 / 1223.61s / 12.78s	0.7801 / 258.70s / 12.21s	0.8879 / 119.36s / 12.97s	0.8966 / 379.59s /12.44s
	BiMax w/Mean	**0.9612** / 71.14s / 0.49s	0.5348‡ / 52.93s / 0.53s	**0.9052**†‡ / 49.09s / 0.51s	**0.9569**† / 49.32s / 0.55s	**0.9138**† / 74.47s / 0.53s	0.7845‡ / 1205.91s / 0.54s	**0.9181**†‡ / 258.35s / 0.58s	**0.9483**‡ / 119.36s / 1.38s	0.9267‡ / 381.05s / 1.04s

Table 6: The results from various sentence embedding models, segmentation strategies, and document alignment methods on the MnRN dataset. where "#Param." represents for the number of parameters, "#Dim." represents for the embedding dimension, "#Lang." represents for the language mode (multilingual or monolingual), "#Arch." represents for the model architecture, "PPROC Time" represents for preprocessing time, "SBS" represents for sentence-based segmentation, "OFLS" represents for overlapping fixed-length segmentation, "FL" represents for fixed-length, "OR" represents for overlapping rate, "Max" represents the token limitation of Blob. Moreover, we put the highest F1 scores achieved by each model under each segmentation strategy in **bold**. For each segmentation strategy within each model, † is appended when BiMax demonstrates statistically significant superiority over both Mean-Pool and TK-PERT, and ‡ is used when it is significantly superior to OT.

Table 7 and Table 8[13]. For the WMT16 test data, we configure OFLS with a sliding-window size of 100 and an overlap ratio of 0.5, the same as Wang et al. (2024c). For the Fernando dataset, we use a fixed-length window of 30 with the same overlap ratio. In our Faiss Search (Johnson et al., 2019) setup, we use IndexFlatIP as the index type and perform cosine similarity searches on GPUs.

	WMT16 test data	MnRN dataset
En Docs.	682k	931
Fr Docs.	522k	-
Ja Docs.	-	232
Web domains	203	4
Gold Pairs	2,402	263
Direction	Fr-En	Ja-En

Table 7: Basic information for the WMT16 test data and MnRN dataset.

Web-domain	No. of Docs.			Aligned Docs.		
	En	Si	Ta	En-Si	En-Ta	Si-Ta
Army	2,081	2,033	1,905	1,848	1,671	1,578
Hiru	1,634	3,133	2,886	1,397	1,056	2,002
ITN	1,942	4,898	1,521	352	112	34
NewsFirst	2,278	1,821	2,333	344	316	97

Table 8: Basic information for the Fernando dataset.

The final document alignment output follows a 1-1 rule (Buck and Koehn, 2016a), whereby each document ID should appear only once in the results. Consequently, we rank all matched document pairs by similarity and eliminate any lower-ranked pairs that contain a document ID already assigned at a higher rank.

For evaluation of the WMT16 document alignment shared task, we adhere to previous work (Buck and Koehn, 2016a; Thompson and Koehn, 2020; Sannigrahi et al., 2023; Wang et al., 2024c) via a "soft" recall metric, which assigns credit to document pairs if either the English or French document (but not both) deviates from the reference document pair by less than 5%, based on text edit distance. For the MnRN dataset, we follow Wang et al. (2024c) in using the F1 score for evaluation. Since multiple correct target documents may correspond to a single source document in MnRN dataset, both precision and recall are calculated with respect to the source-side instances within the set of gold pairs (i.e., although there are 263 gold pairs, they involve only 232 unique source documents; therefore, we define the total number of instances as 232.)

[13]We use the data provided by the authors on GitHub, whose size differs from that reported in the original paper

For significance testing, we refer to Yeh (2000) and adopt a randomization test procedure. Given two result sets, A and B, with corresponding F1 scores F1_A and F1_B (assuming $\text{F1}_A > \text{F1}_B$), we retain their intersection $A \cap B$ and isolate the symmetric difference $A \triangle B$, irrespective of the correctness of each pair. The elements in $A \triangle B$ are then randomly partitioned into two subsets, yielding $2^{|A \triangle B|}$ possible permutations. For each trial, the two subsets are combined with $A \cap B$ to form new result sets A' and B', from which updated F1 scores $\text{F1}_{A'}$ and $\text{F1}_{B'}$ are computed. Let n denote the number of trials in which $(\text{F1}_{A'} - \text{F1}_{B'} > \text{F1}_A - \text{F1}_B)$; the p-value is then calculated as $(n+1)/(2^{|A \triangle B|}+1)$. Following Yeh (2000), when $|A \triangle B| > 20$, we use an approximate randomization with 1,048,576 shuffles.

All experiments are conducted on two A6000 GPUs and one H100 GPU.

D Discussion of Blob and its Variants

We select four well-performing models in Section 4.1 to investigate the token limitation of Blob. The results are presented in Table 9.

As the token limitation increases, the number of blobs segmented from the document decreases accordingly, resulting in a natural reduction in embedding computation time. Furthermore, in most cases, the highest F1 Score is achieved with a token limitation of 64. Therefore, prioritizing accuracy, we compare the results in this case with SBS and OFLS in Section 4.1.

Strategies & Models		Embedding Models			
		LaBSE	distiluse-base-multi-cased-v2	BGE M3 (dense only)	jina-embeddings-v3
Experiments (F1 Score ↑ / PPROC. Time (sec.) ↓)					
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s	**0.8663** / 70.80s	0.8750 / 565.45s	**0.8448** / 127.75s
	TK-PERT	**0.8663** / 164.87s	**0.8491** / 139.41s	0.8534 / 655.54	0.8578 / 220.72s
	OT w/Mean	**0.8233** / 107.84s	**0.8405** / 70.46s	**0.8362** / 564.84s	**0.8276** / 128.12s
	BiMax w/Mean	0.9009 / 106.65s	**0.9052** / 71.16s	0.9181 / 564.76s	**0.9052** / 127.32s
Blob (Max 128)	Mean-Pool	**0.8879** / 75.65s	0.8233 / 52.60s	0.8491 / 581.87s	**0.8448** / 110.10s
	TK-PERT	0.8621 / 109.81s	**0.8491** / 94.63s	0.8664 / 640.70s	**0.8707** / 169.36s
	OT w/Mean	0.8190 / 75.74s	0.8103 / 52.73s	0.8017 / 582.46s	0.8060 / 108.08s
	BiMax w/Mean	**0.9138** / 75.98s	0.8621 / 52.69s	**0.9267** / 581.77s	0.9009 / 108.94s
Blob (Max 256)	Mean-Pool	0.8793 / 52.09s	0.8362 / 38.53s	0.8491 / 475.60s	0.8147 / 99.09s
	TK-PERT	0.8491 / 74.41s	0.8405 / 64.89s	0.8578 / 510.18s	0.8276 / 136.84s
	OT w/Mean	0.7543 / 51.96s	0.7457 / 38.70s	0.7629 / 475.64s	0.7457 / 99.26s
	BiMax w/Mean	0.8879 / 51.97s	0.8879 / 38.48s	**0.9267** / 475.76s	0.8966 / 99.15s
Blob (Max 384)	Mean-Pool	0.8706 / **45.00s**	0.8362 / **34.21s**	**0.9138** / **424.74s**	0.8190 / **95.54s**
	TK-PERT	0.8147 / **62.72s**	0.7802 / **54.38s**	**0.8879** / **466.63s**	0.8621 / **125.52s**
	OT w/Mean	0.6552 / **44.97s**	0.6336 / **34.41s**	0.6853 / **424.56s**	0.6638 / **95.42s**
	BiMax w/Mean	0.8793 / **45.09s**	0.8232 / **34.30s**	0.8879 / **424.34s**	0.8966 / **95.79s**

Table 9: The results of different max token limitation settings for Blob. Each model's highest F1 scores and shortest preprocessing time are tagged in **bold**.

Based on the ablation analysis conducted by Wang et al. (2024c), the overlapping rate has a notable impact on the accuracy of OFLS. Therefore, we hypothesize that appropriately introducing overlapping parts between Blobs might contribute to improvement. We design the following three

Strategies & Models		Embedding Models			
		(a) LaBSE	(b) distiluse-base-multi-cased-v2	(c) BGE M3 (dense only)	(d) jina-embeddings-v3
Experiments of Blob-o w/tok (F1 Score ↑ / PPROC. Time (sec.) ↓)					
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s	0.8663 / 70.80s	**0.8750** / **565.45s**	0.8448 / 127.75s
	TK-PERT	0.8663 / 164.87s	0.8491 / 139.41s	0.8534 / **655.54s**	0.8578 / 220.72s
	OT w/Mean	0.8233 / 107.84s	**0.8405** / 70.46s	**0.8362** / **564.84s**	**0.8276** / 128.12s
	BiMax w/Mean	0.9009 / 106.65s	**0.9052** / 71.16s	0.9181 / **564.76s**	0.9052 / 127.32s
Blob-o w/tok (64, 0.15)	Mean-Pool	**0.9224** / 115.24s	**0.8707** / 77.39s	0.8448 / 594.08s	**0.8534** / 136.96s
	TK-PERT	**0.8966** / 178.46s	0.8664 / 153.17s	0.8664 / 686.60s	**0.8922** / 231.78s
	OT w/Mean	**0.8793** / 115.38s	0.8147 / 77.26s	**0.8362** / 593.45s	**0.8276** / 137.20s
	BiMax w/Mean	0.9181 / 114.95s	0.8922 / 77.14s	0.9095 / 594.40s	**0.9267** / 137.36s
Blob-o w/tok (128, 0.15)	Mean-Pool	0.8966 / **82.35s**	0.8491 / **59.84s**	0.8707 / 599.33s	0.8491 / **118.38s**
	TK-PERT	0.8879 / **124.02s**	**0.8707** / **110.24s**	**0.8750** / 672.73s	0.8578 / **182.58s**
	OT w/Mean	0.8448 / **82.69s**	0.8017 / **59.86s**	0.7844 / 600.28s	0.7931 / **119.64s**
	BiMax w/Mean	**0.9353** / **82.90s**	0.8793 / **59.91s**	**0.9310** / 598.58s	**0.9267** / **134.87s**
Experiments of Blob-o w/sent (F1 Score ↑ / PPROC. Time (sec.) ↓)					
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s	**0.8663** / 70.80s	**0.8750** / **565.45s**	**0.8448** / 127.75s
	TK-PERT	0.8663 / 164.87s	0.8491 / 139.41s	0.8534 / 655.54	0.8578 / 220.72s
	OT w/Mean	**0.8233** / 107.84s	**0.8405** / 70.46s	0.8362 / **564.84s**	**0.8276** / 128.12s
	BiMax w/Mean	0.9009 / 106.65s	**0.9052** / 71.16s	0.9181 / **564.76s**	**0.9052** / 127.32s
Blob-o w/sent (64, 4)	Mean-Pool	0.8707 / 108.21s	0.8621 / 73.74s	**0.8750** / 600.28s	0.8405 / 130.19s
	TK-PERT	**0.8879** / 173.63s	**0.8578** / 149.27s	**0.8621** / 655.37s	**0.8664** / 223.978s
	OT w/Mean	**0.8233** / 108.82s	0.8362 / 73.87s	**0.8405** / 600.49s	**0.8276** / 129.78s
	BiMax w/Mean	0.9009 / 108.56s	0.9009 / 73.63s	0.9138 / 600.78s	**0.9052** / 130.08s
Blob-o w/sent (128, 3)	Mean-Pool	**0.8966** / **84.83s**	0.8189 / **59.92s**	0.8621 / 578.21s	0.8319 / **116.21s**
	TK-PERT	**0.8879** / **124.49s**	0.8362 / **108.65s**	0.8534 / **638.59s**	0.8578 / **179.93s**
	OT w/Mean	0.7974 / **85.09s**	0.7931 / **60.84s**	0.7931 / 577.49s	0.8060 / **116.06s**
	BiMax w/Mean	**0.9095** / **84.78s**	0.8707 / **60.23s**	**0.9224** / 577.78s	0.8966 / **116.61s**
Experiments of Blob-o w/tok-lim (F1 Score ↑ / PPROC. Time (sec.) ↓)					
Blob (Max 64)	Mean-Pool	0.8621 / 107.02s	**0.8663** / 70.80s	**0.8750** / **565.45s**	**0.8448** / 127.75s
	TK-PERT	0.8663 / 164.87s	0.8491 / 139.41s	0.8534 / **655.54**	0.8578 / 220.72s
	OT w/Mean	**0.8233** / 107.84s	**0.8405** / 70.46s	**0.8362** / **564.84s**	0.8276 / 128.12s
	BiMax w/Mean	0.9009 / 106.65s	**0.9052** / 71.16s	0.9181 / **564.76s**	**0.9052** / 127.32s
Blob-o w/tok-lim (64, 0.15)	Mean-Pool	0.8621 / 111.63s	**0.8663** / 74.65s	**0.8750** / 613.27s	0.8405 / 130.53s
	TK-PERT	**0.8793** / 175.39s	**0.8578** / 148.23s	**0.8578** / 672.91s	**0.8621** / 225.43s
	OT w/Mean	**0.8233** / 111.41s	**0.8405** / 74.97s	**0.8362** / 613.65s	**0.8319** / 129.22s
	BiMax w/Mean	0.9009 / 111.46s	**0.9052** / 75.51s	**0.9267** / 613.19s	0.9009 / 129.78s
Blob-o w/tok-lim (128, 0.45)	Mean-Pool	**0.8966** / **81.94s**	0.8319 / **58.93s**	0.8578 / 593.52s	0.8103 / **117.29s**
	TK-PERT	0.8707 / **122.81s**	0.8491 / **107.59s**	0.8491 / **654.17s**	0.8448 / **179.66s**
	OT w/Mean	0.7974 / **82.56s**	0.7672 / **58.97s**	0.7931 / 593.12s	0.8017 / **117.22s**
	BiMax w/Mean	**0.9138** / **82.15s**	0.8836 / **58.80s**	**0.9267** / 593.23s	0.8750 / **116.90s**

Table 10: The comparative results between the three Blob-o approaches and the original Blob method. Each model's highest F1 scores and shortest preprocessing time are tagged in **bold** with underline.

approaches:

Blob-o w/tok: For any two given Blobs A and B and a specified ratio r, we copy the last $len(A) \times r$ tokens from Blob A to the beginning of Blob B while simultaneously replicating the first $len(B) \times r$ tokens from Blob B to the end of Blob A, with all operations performed at the token level.

Blob-o w/sent: For any two given Blobs A and B and a specified number n, we copy the last n sentences from Blob A to the beginning of Blob B while simultaneously replicating the first n sentences from Blob B to the end of Blob A, with all operations performed at the sentence level.

Blob-o w/tok-lim: For any two given Blobs A and B and a specified ratio r, we copy the multiple sentences from the end of Blob A, comprising no more than $len(A) \times r$ tokens, to the beginning of Blob B, while simultaneously replicating multiple sentences from the beginning of Blob B, containing no more than $len(B) \times r$ tokens, to the end of Blob A, with all operations performed at the sentence level.

We perform preliminary experiments using the Mean-Pool method based on the LaBSE model to examine appropriate combinations of the maximum token limitation for Blob composition and the hyperparameters associated with the aforementioned

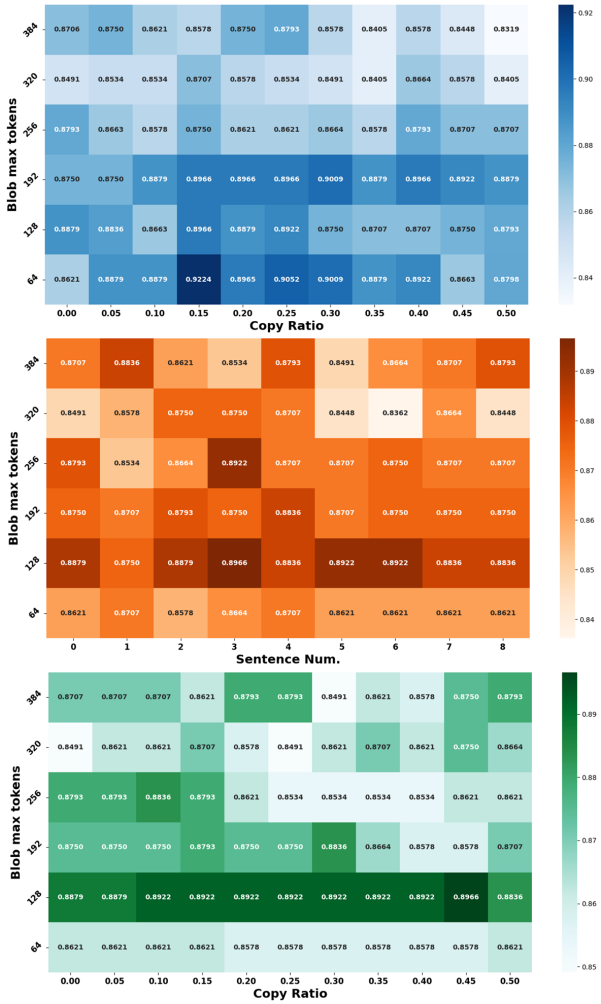

Figure 2: The preliminary experimental results of the three Blob-o approaches using the Mean-Pool method based on the LaBSE model. From top to bottom, the results correspond to Blob-o w/tok, Blob-o w/sent, and Blob-o w/tok-lim, respectively.

three approaches. The results of these experiments are presented in Figure 2.

The exploration is limited to the LaBSE model and Mean-Pool method, possibly creating bias if their optimal configurations are directly applied to alternative models and document alignment methods. To address this, we implement a comparative evaluation by examining two cases from each Blob-o approach against the original Blob method.

As shown in Table 10, except for the Blob-o w/tok (64, 0.15) with the LaBSE model, which performs well compared to the original Blob (Max 64), other cases do not exhibit significant improvement. Combined with the results in Table 9, it can be observed that the OT method does not integrate well with Blob, potentially because OT tends to achieve superior performance with finer segmentation granularity, which contradicts the Blob approach.

The Blob method is initially introduced to our research to reduce the impact of boilerplate text, preserve the meaning of source sentences, and accelerate embedding processes. However, on the MnRN Dataset, it shows little improvement compared to SBS. This could be attributed to multiple factors, such as the embedding model's potential inability to effectively represent long segments, the relatively small scale of the MnRN dataset, and the lack of coherence between Blobs. Nevertheless, the results indicate that even with the use of overlapping, the performance of the Blob method on the MnRN dataset has not been enhanced overall.

E Detailed experimental results on the Fernando Dataset

The detailed experimental results are reported in Table 11. Following Fernando et al. (2023), we record the precision, recall, and F1 score of each language pair under each web domain. Since only the Army and Hiru domains in the GitHub data provided by the authors match the data size reported in the original paper, we reproduce their strong baseline method "GMD-SL" (i.e., the GMD method with segment length as the weighting scheme) using the authors' released code[14] for comparison. The ITN and NewsFirst domains are excluded from comparison due to substantial discrepancies in the data. Moreover, since OT in this paper adopts segment frequency as its weighting strategy (OT-SF), we include the results of "OT-SL + SBS" to ensure a fair comparison with GMD.

As shown in Table 11, even on the Army and Hiru domains, the results of GMD-SL reported in the original paper differ substantially from those reproduced in our experiments. This discrepancy is the main reason we do not compare our results directly with the original paper. Then, BiMax consistently achieves the highest accuracy in all cases. However, it can be observed that TK-PERT and OT, which performed well on the WMT16 test data, do not achieve satisfactory results in this experiment. This may be attributed to the fact that the Fernando dataset primarily consists of relatively short documents, whereas both methods are better suited for long texts. Moreover, this also reveals a limitation of the current version of TK-PERT: the

[14] https://github.com/nlpcuom/parallel_corpus_mining/blob/master/document_alignment/GreedyMoversDistance.py

LaBSE	Segment Strategy	Army								
		En-Si			En-Ta			Si-Ta		
		R	P	F1	R	P	F1	R	P	F1
Mean-Pool	SBS	98.16	90.47	94.16	80.08	81.52	80.79	81.88	75.03	78.30
	OFLS	98.43	90.27	94.18	93.83	85.17	89.29	94.04	79.58	86.20
TK-PERT	SBS	98.32	90.53	94.27	84.20	81.51	82.83	86.63	76.07	81.01
	OFLS	98.81	90.85	94.66	94.36	85.97	89.97	95.18	81.54	87.84
OT-SF	SBS	98.05	90.46	94.11	83.49	84.38	83.93	85.36	77.73	81.37
	OFLS	98.97	90.81	94.72	96.24	87.17	91.48	97.34	82.32	89.20
BiMax	SBS	98.32	90.71	94.37	85.49	86.30	85.89	87.58	79.75	83.48
	OFLS	**99.30**	**91.12**	**95.03**	**97.46**	**88.14**	**92.52**	**97.72**	**82.48**	**89.50**
GMD-SL (orig)	SBS	99.73	94.85	97.23	98.47	91.89	95.07	99.11	86.84	92.57
GMD-SL (our)	SBS	98.43	90.72	94.42	86.02	87.83	86.42	88.59	80.62	84.42
	OFLS	**99.30**	91.02	94.98	96.89	87.67	92.05	97.47	82.33	89.26
OT-SL	SBS	98.27	90.53	94.24	84.78	85.69	85.23	87.52	79.73	83.44

LaBSE	Segment Strategy	Hiru								
		En-Si			En-Ta			Si-Ta		
		R	P	F1	R	P	F1	R	P	F1
Mean-Pool	SBS	89.33	76.38	82.35	73.59	52.69	61.41	77.02	57.47	65.83
	OFLS	88.69	75.83	81.76	75.81	54.28	63.27	77.87	57.93	66.44
TK-PERT	SBS	80.46	68.79	74.17	58.89	42.24	49.20	66.18	49.85	56.87
	OFLS	82.03	70.13	75.62	65.90	47.18	54.99	69.73	51.84	59.47
OT-SF	SBS	86.75	74.17	79.97	70.60	50.55	58.92	77.62	57.81	66.27
	OFLS	86.75	74.17	79.97	71.37	51.10	59.56	78.27	58.21	66.77
BiMax	SBS	**93.49**	**79.93**	**86.18**	79.74	57.13	66.57	82.42	60.93	70.06
	OFLS	93.13	79.62	85.85	**82.48**	**59.06**	**68.83**	**83.27**	**61.38**	**70.67**
GMD-SL (orig)	SBS	95.42	82.44	88.45	87.09	62.71	72.92	87.46	65.19	74.66
GMD-SL (our)	SBS	91.98	78.64	84.79	78.12	55.94	65.19	80.42	59.39	68.32
	OFLS	89.05	76.13	82.09	78.55	56.24	65.66	79.87	58.87	67.78
OT-SL	SBS	88.33	75.52	81.43	72.22	51.71	60.27	78.12	58.23	66.72

LaBSE	Segment Strategy	ITN								
		En-Si			En-Ta			Si-Ta		
		R	P	F1	R	P	F1	R	P	F1
Mean-Pool	SBS	85.51	15.69	26.52	74.11	5.98	11.07	91.18	2.05	4.02
	OFLS	83.24	15.47	26.09	83.04	6.78	12.54	91.18	2.04	4.00
TK-PERT	SBS	78.69	14.40	24.34	72.32	5.80	10.74	85.29	1.92	3.75
	OFLS	80.40	14.81	25.01	79.46	6.36	11.78	79.41	1.78	3.48
OT-SF	SBS	82.67	15.15	25.60	76.79	6.23	11.53	91.18	2.06	4.02
	OFLS	83.52	15.50	26.14	83.04	6.82	12.60	88.24	1.98	3.87
BiMax	SBS	**89.49**	**16.35**	**27.64**	83.04	6.71	12.42	**97.06**	**2.19**	**4.28**
	OFLS	86.36	15.97	26.96	**91.96**	**7.44**	**13.76**	91.18	2.04	4.00
GMD-SL (our)	SBS	88.07	16.09	27.20	85.71	6.99	12.92	94.12	2.12	4.14
	OFLS	85.80	15.88	26.80	83.93	6.83	12.63	88.24	1.98	3.87
OT-SL	SBS	83.24	15.25	25.78	75.89	6.15	11.39	94.12	2.12	4.14

LaBSE	Segment Strategy	NewsFirst								
		En-Si			En-Ta			Si-Ta		
		R	P	F1	R	P	F1	R	P	F1
Mean-Pool	SBS	88.95	18.29	30.34	81.33	12.93	22.31	84.54	4.67	8.85
	OFLS	90.99	18.17	30.28	86.71	13.54	23.43	87.63	4.82	9.14
TK-PERT	SBS	87.79	17.71	29.48	81.01	12.84	22.17	82.47	4.53	8.58
	OFLS	86.92	17.27	28.82	82.59	12.84	22.22	82.47	4.53	8.60
OT-SF	SBS	88.37	18.16	30.13	86.08	13.57	23.45	85.57	4.75	9.00
	OFLS	90.41	18.06	30.11	87.34	13.50	23.39	88.66	4.88	9.25
BiMax	SBS	**95.06**	19.34	**32.14**	90.51	**14.25**	**24.62**	88.66	4.88	9.25
	OFLS	93.02	**19.84**	30.74	**91.46**	14.13	24.47	**91.75**	**4.98**	**9.44**
GMD-SL (our)	SBS	93.02	18.95	31.48	90.19	14.20	25.54	88.66	4.87	9.23
	OFLS	93.90	18.55	30.98	**91.46**	14.03	24.33	89.69	4.88	9.26
OT-SL	SBS	89.53	18.38	30.50	86.71	13.67	23.62	86.60	4.78	9.06

Table 11: The detailed results on the Fernando dataset. We highlight the best one in **bold** for each column. "GMD-SL (orig)" represents the results in the original paper (Fernando et al., 2023).

number of windows per document is fixed, which is clearly suboptimal for handling datasets with diverse length distributions, along with the challenge of properly configuring hyperparameters in advance. These inabilities remain an issue that TK-PERT could potentially improve upon.

As a greedy-search variant of OT, GMD achieves better performance than OT on the Fernando dataset, ranking second only to BiMax and clearly demonstrating its superior accuracy. However, due to its exhaustive traversal of all segment pairs, the computational cost grows significantly with the number of segments, leading to a marked slowdown, particularly in the case that OFLS divides documents into shorter segments. When running GMD using the implementation provided by Fernando et al. (2023), its throughput under the OFLS setting was more than 20 times lower than BiMax and even more than 5 times lower than OT, indicating that further improvements are needed at both the algorithmic and implementation levels.

F Alignment Accuracy Analysis based on Document Length

Since English is the common language across the MnRN dataset, WMT16 test data, and the Fernando dataset[15], we examine the length distribution of the English documents in the gold data, measuring text length by the number of tokens obtained after tokenization with the LaBSE model. Figure 3 presents the recall performance of the OFLS-based methods across different length intervals.

It can be observed that the length distributions of the gold data differ across the three datasets. The MnRN dataset exhibits a relatively balanced distribution but contains more short documents; the WMT16 test data are primarily concentrated on long documents; in contrast, the Fernando dataset is almost entirely composed of short texts, with virtually no long documents.

Overall, embedding-based document alignment methods tend to perform less effectively on short texts, particularly for documents in the $[0, 256)$ length interval. Among them, BiMax demonstrates the strongest capability in handling short documents, achieving a recall of 0.95 in the $[0, 256)$ interval of the Fernando dataset. Nevertheless, this performance is still lower than the accuracy achieved by TK-PERT and OT on long documents in the $[2048, \infty)$ interval of the WMT16 test data. A potential reason for this, as revealed through our examination of the MnRN dataset, is that even short documents often contain a substantial amount of boilerplate text, which appears repeatedly across documents within the same web domain. This repetition further reduces the space for discriminative content, making it difficult for sentence-level embedding methods to capture fine-grained features.

As we noted in Section 6 and Appendix E, the strong performance of TK-PERT and OT on the WMT16 test data is largely attributable to their effectiveness in handling long documents, an aspect where BiMax is comparatively weaker. Conversely, these two methods perform less effectively than BiMax on the short-text Fernando dataset. This suggests that, rather than relying on a single alignment method, it may be worthwhile to consider a system that leverages different alignment approaches within the length intervals where each performs best.

G Downstream MT work for Document Alignment on the WMT23 Data Task

Up to date, there exist several datasets for evaluating document alignment tasks (e.g., WMT16 document alignment task (Buck and Koehn, 2016a), CC-Aligned Dataset (El-Kishky et al., 2020)). However, the evaluations typically measure the accuracy of document alignment methods using recall or F1 scores on document pairs. There has not yet been a publicly available system that evaluates document alignment accuracy through machine translation performance as a downstream task.

The WMT23 parallel data curation shared task (WMT23 data task) (Sloto et al., 2023) focuses on identifying the best MT training data from provided web-crawled data, including both document and sentence levels. The final developed datasets are evaluated using a unified end-to-end MT system. As one of the participants, Steingrimsson (2023) employed the document alignment method for one part of dataset creation, ultimately combining it with the dataset via multi-filtering techniques to produce the final dataset. However, he did not explore their document alignment methodology in depth.

We hope to develop a comparative benchmark on the Estonian-Lithuanian (et-lt) WMT23 data task and establish an end-to-end system for the document alignment task that utilizes machine transla-

[15]For the Fernando dataset, we merge the En–Si data from the four web-domains for our analysis.

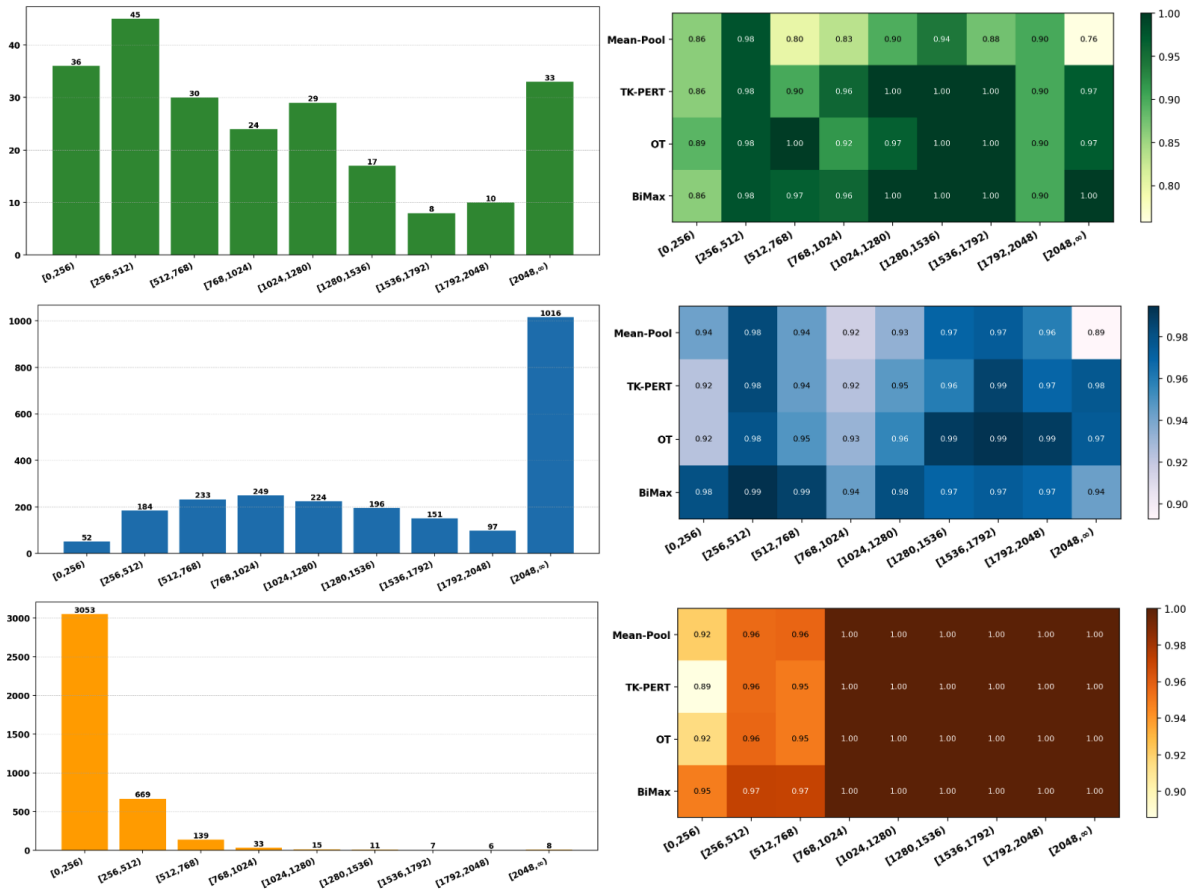

Figure 3: The length distribution of the English side of the gold data and the recall results of OFLS-based document alignment methods across different length intervals. From top to bottom, the results correspond to the MnRN dataset, WMT16 test data, and Fernando dataset, respectively.

tion (MT) accuracy as the evaluation metric. However, the process of converting document-aligned data into a parallel corpus involves multiple steps, such as sentence alignment and sentence pair filtering. It can be anticipated that these steps will increase the permissible error margin for document alignment methods, which means that when different document alignment methods achieve sufficiently high accuracy, the resulting datasets may not exhibit significant differences in quality.

G.1 Procedure for Hierarchical Data Curation

G.1.1 Preprocessing with CH Data

First, since documents from different hostnames (web domains) are unlikely to be translations of each other, we extract common hostnames that appear in both [*documents.et.tsv*] and [*documents.lt.tsv*] files provided by the WMT23 data task. We then perform the following two preprocessing steps: (1) Since documents may contain the same content even with different document IDs (docids), in order to conserve computational and storage resources during the subsequent embedding process and to reduce redundant sentence pairs in the final parallel dataset, we deduplicate the source and target documents, respectively, within each hostname. While cross-hostname duplicates also exist, we restrict deduplication to within-hostname operations to prevent certain hostnames from being completely depleted of documents. (2) We remove exceptionally long documents[16], specifically, those whose length exceeds ten times the maximum length of documents in the opposed language, which can reasonably be assumed to lack aligned counterparts. Following the steps described above, we divide the resulting Common Hostname Data (CH Data) into two categories:

- Common Hostname Data 1 (CH Data 1): Hostname data that have only one document

[16]For instance, for the hostname "lt.airbnb.com", under LaBSE tokenization, since the longest document on the Estonian side does not exceed 1,000 tokens, we remove documents from the Lithuanian side that contain more than 10,000 tokens.

on both the Estonian and Lithuanian sides.

- Common Hostname Data 2 (CH Data 2): Hostname data for which at least one side (et or lt) contains multiple documents.

We collect some information about the CH Data and record it in Table 12. Each document is stored in the format "URL\t Hostname\t docid\t content (encoded in base64)".

	CH Data 1	CH Data 2	CH Data
Hostname num.	6,791	17,529	24,320
Estonian Docs.	6,791	419,152	425,943
Lithuanian Docs.	6,791	393,742	400,533

Table 12: Some statistical information of CH Data.

G.1.2 Document alignment

Next, we perform document alignment between the et-lt documents. We directly compute the similarity for CH Data 1 since each hostname can contain only a single document pair, and for CH Data 2, we perform retrieval. The final training and evaluation scripts provided by the WMT23 data task focus on the et-lt direction, so we follow them to set the retrieval direction as et-lt. Subsequently, we merge the results from the two document alignment processes to obtain the CH document pairs (CH docpairs) and the similarity score for each pair.

G.1.3 Document-level Filtering

Because we perform deduplication only within each hostname for CH Data, the resulting CH docpairs may still contain repeated content, and we cannot fully ensure that all documents are genuinely in Estonian or Lithuanian. Hence, we apply document-level filtering as follows:

(I) Deduplication: We sort the CH docpairs by similarity scores in descending order. If an Estonian or Lithuanian document reappears in a later pair, we remove that occurrence. In other words, we retain only the pairing with the highest similarity score for each document to eliminate duplicates.

(II) Language identification: Using the FastText model[17] (Joulin et al., 2016a,b), we identify the language of each document in the remaining pairs from (I). We only preserve pairs whose source document is most likely Estonian (et) and whose target document is most likely Lithuanian (lt).

[17] https://fasttext.cc/docs/en/language-identification.html

Since the number of docpairs developed from different document alignment methods varies, with the goal of comparing these methods, from a fairness standpoint, a fair comparison would require setting a similarity threshold or fixing the sampling size to extract docpairs. However, because the similarity scales produced by methods (e.g., OT and BiMax) differ substantially, adopting a single fixed threshold is impossible. Consequently, we select a specified number of top-ranked docpairs (based on similarity) from the document-level filtering results for the subsequent sentence alignment.

G.1.4 Sentence Alignment

We use Vecalign (Thompson and Koehn, 2019) to perform sentence alignment on the docpairs obtained in Section G.1.3. Differing from the default settings, we set the overlap to 4 and replace the embedding from LASER to LaBSE. Furthermore, using each sentence's index in the document, we find the corresponding sentence ID (sentid) in the file we compiled, which is limited to the Common Hostname part from the [*sentences.et.tsv*] and [*sentences.lt.tsv*] provided by the WMT23 data task.

G.1.5 Sentence-level Filtering

In this step, we do not propose or employ any novel or complex methodology. Instead, we carry out the necessary removal with the test and development data, as well as quality-based filtering of sentence pairs (sentpairs):

(III)Test&Dev Removal: Relying on the organizer-provided [*exclude_sent_ids_et-lt.txt*], we remove all sentpairs whose sentid covers with any ID listed in this file.

(IV) Quality-Based Filtering: Following the approach of Steingrimsson (Steingrimsson, 2023), we retain only those pairs in which both the Estonian and Lithuanian sentences have more than three tokens (tokenized simply by space). We then use the LaBSE model for embedding and compute the cosine similarity for each sentpair, removing any pairs with a score below 0.4.

Similarly to Section G.1.3, we sort the filtered sentpairs in descending order of cosine similarity and extract a fixed number of pairs as our final parallel dataset for training.

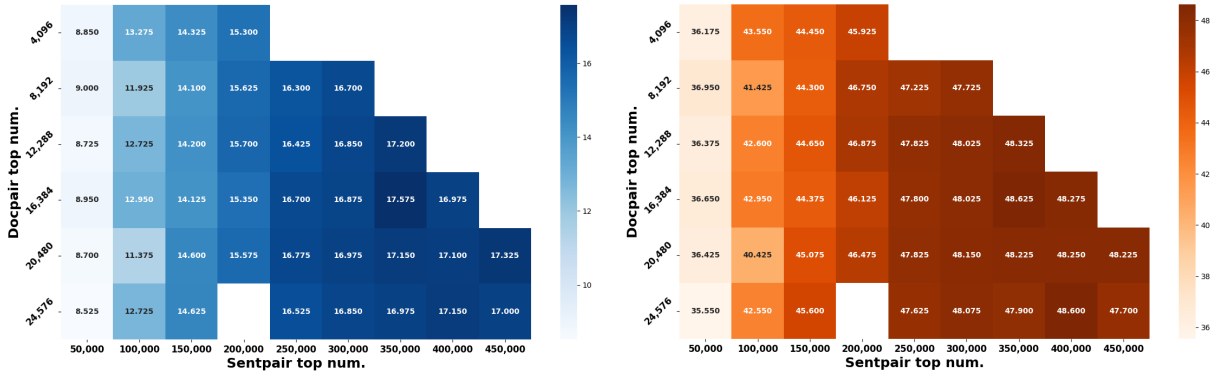

Figure 4: Results of Avg BLEU (left) and Avg ChrF (right) for Mean-Pool with OFLS (40, 0.5).

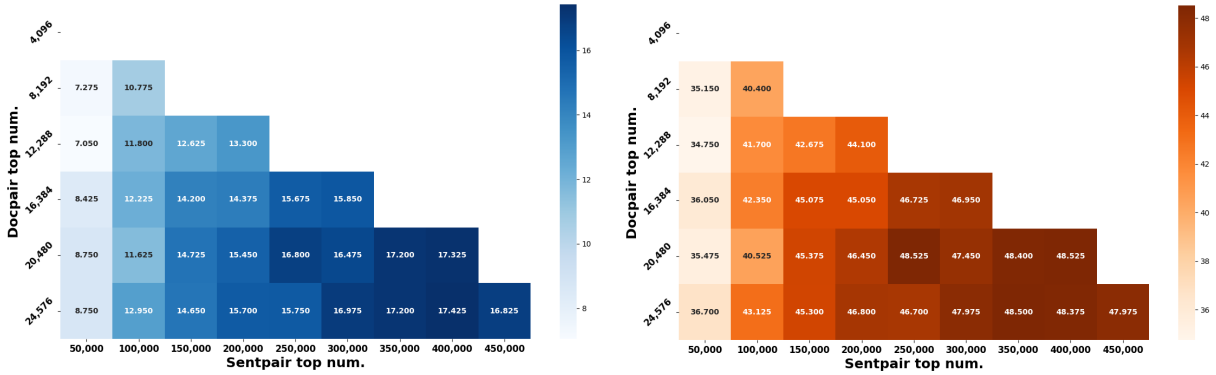

Figure 5: Results of Avg BLEU (left) and Avg ChrF (right) for OT w/Mean with OFLS (40, 0.5).

G.2 Downstream MT Benchmark

G.2.1 Experiment Settings

As in our experiments on the WMT16 document alignment task and the MnRN dataset, we employ four document alignment methods, Mean-Pool, TK-PERT, OT, and BiMax, as well as two segmentation strategies, SBS and OFLS. The difference is that we rely solely on Mean-Pool to retrieve candidates for OT and BiMax, and we set the fixed-length to 40 and the overlapping rate to 0.5 for OFLS (OFLS (40, 0.5)).

We use the provided scripts to conduct end-to-end training, applying an early-stopping criterion that terminates training if the validation perplexity (PPL) does not improve for 12 consecutive epochs. Then, the checkpoint with the lowest PPL is selected as the best model. However, the procedure for model selection will be discussed in more detail in Section G.2.3.

The evaluation uses BLEU (Papineni et al., 2002) and chrF (Popović, 2015) across four domains[18] : EMEA, EUbookshop (EUB), Europarl (EP), and JRC-Acquis (JRC). All datasets are released by OPUS[19] (Tiedemann, 2012).

We conduct all experiments except training on two H100 GPUs, while the NMT model training is done on one A6000 GPU[20].

G.2.2 Selection for Docpairs and Sentpairs

Before conducting experiments on all combinations of document alignment methods and segmentation strategies, we should determine the number of docpairs and sentpairs required for our benchmark, as described in Sections G.1.3 and G.1.5. Therefore, we select Mean-Pool and OT with the OFLS (40, 0.5) as representatives, gradually reducing the number of docpairs from 24,576 in increments of 4,096, and running experiments in the step of 50,000 sentpairs for each docpair number. Note that after document-level filtering, although Mean-Pool and OT with OFLS (40, 0.5) retain 34,173 and 33,802

[18]The organizers add EUconst as an additional held-out domain in the Findings paper (Sloto et al., 2023) as part of the held-out test set, which has not been publicly available.

[19]https://opus.nlpl.eu/

[20]At the time this study was finished, unfortunately, the Sockeye Python library used by the training script was not yet compatible with Torch version 2.0 or higher, which is necessary for H100 GPUs. Thus, we migrated model training to one A6000 GPU. However, because the original script assumes the use of eight V100 GPUs, we increased the batch size to eight times its original size.

Document Alignment	Segmentation Strategy	Sim. Time (sec.)		Pair num.		
		CH Data 1	CH Data 2	Orig. Docpairs	Doc-Flt Docpairs	Vecalign (top 24,586 docpairs) + Sent-Flt Sentpairs
Mean-Pool	SBS	112.03s	10148.23s	78,539	33,627 ⇒ 24,586	493,024 → 400,000
	OFLS (40, 0.5)	94.55s	8043.31s	79,362	34,173 ⇒ 24,586	490,693 → 400,000
TK-PERT	SBS	125.25s	10646.20s	78,785	33,722 ⇒ 24,586	501,745 → 400,000
	OFLS (40, 0.5)	107.43s	8750.86s	79,405	34,138 ⇒ 24,586	496,471 → 400,000
OT w/Mean	SBS	177.32s	25761.43s	78,655	33,306 ⇒ 24,586	491,069 → 400,000
	OFLS (40, 0.5)	144.86s	20416.14s	79,466	33,802 ⇒ 24,586	483,219 → 400,000
BiMax w/Mean	SBS	143.13s	13018.73s	78,694	33,818 ⇒ 24,586	456,492→ 400,000
	OFLS (40, 0.5)	120.34s	10472.15s	79,463	34,228 ⇒ 24,586	455,765 → 400,000

Table 13: Some information on the data development process, "Sim. Time" represents the time cost by the similarity calculation, "Orig. Docpairs" represents the docpairs before document-level filtering, "Doc-Flt Docpairs" represents the docpairs after document-level filtering, "Vecalign (top 24,586 docpairs) + Sent-Flt Sentpairs" represents the sentpairs after sentence-level filtering using top 24,586 docpairs.

pairs, respectively, at the 24,576 point, the docpair similarities for Mean-Pool and OT reach 0.75 and 0.32 (in the range [0,1]), with OT's similarity value already considered very low.

Since presenting the BLEU and ChrF scores separately for all four domains would yield an enormous volume of data and complicate visualization, we use the average BLEU (avg BLEU) and average ChrF (avg ChrF) across the four domains, and the results are shown in Figure 4 and 5.

As shown in Figure 4, at the point of docpair top num. 24,576 and sentpair top num. 200,000, Mean-Pool with OFLS (40, 0.5) yields no data because training failed (possibly due to overfitting). Comparing the two document alignment methods reveals that Mean-Pool provides more sentpairs than OT. Meanwhile, for both methods, regardless of the docpair count, avg BLEU and avg ChrF display a clear upward trend until the sentpair top number reaches 350,000, after which they level off. Therefore, we decide to use **docpair top num. 24,576** and **sentpair top num. 400,000** as our benchmark settings for two reasons as follows:

- Mean-Pool and OT methods roughly achieve their highest accuracy at around 400,000 sentpairs.
- Considering methods like OT that probably generate fewer sentence pairs, we avoid choosing the maximum possible number of sentpairs for each docpair top number (e.g., docpair top num. 24,576 and sentpair top num. 450,000).

G.2.3 Developing MT Benchmark for various method combinations

In this section, we adopt docpair top num. 24,576 and sentpair top num. 400,000, which are determined in Section G.2.2, and follow the hierarchical data curation procedures described in Section G.1. Under these settings, we conduct experiments on all combinations of the four document alignment methods and the two segmentation strategies, and record some details of the data development process in Table 13.

It is apparent that as the number of docpairs increases, compared to the small-scale MnRN dataset, the similarity calculation time gap grows significantly. Consequently, under both SBS and OFLS conditions for CH Data 2, OT takes nearly twice as long as BiMax, whereas Mean-Pool is still the fastest method.

Next, we performed five replicate experiments for each of the methods that produced 400,000 sentpairs in Table 13, and the results are presented in Table 14. However, we do not rely on the original approach of selecting the best checkpoint solely based on the validation perplexity (PPL). We observe that when the dataset size is relatively small, the valid PPL converges more quickly than other metrics, such as BLEU, ChrF, and Rouge-L, indicating that the checkpoint selected exclusively by PPL is unreasonable. Therefore, instead of relying on PPL alone, we sum the rankings for PPL, BLEU, ChrF, and Rouge-L on the validation data to determine the best checkpoint. In cases where multiple checkpoints yield the same total ranking, actually any of those checkpoints can be chosen. Nonetheless, we impose a priority order of BLEU > ChrF > PPL > Rouge-L to select the final best model, and this selection approach is determined as Auto-Rank.

As shown in Table 14, considering both BLEU and ChrF, there are no substantial differences among the document alignment methods or be-

Document Alignment	Segmentation Strategy	BLEU				ChrF			
		EMEA	EUB	EP	JRC	EMEA	EUB	EP	JRC
Mean-Pool	SBS	14.7±0.1	17.7±0.1	15.7±0.2	23.9±0.1	44.3±0.1	50.4±0.1	49.0±0.1	53.6±0.2
	OFLS (40, 0.5)	14.7±0.2	17.8±0.2	15.7±0.2	23.9±0.1	44.4±0.3	50.6±0.2	49.0±0.2	53.7±0.2
TK-PERT	SBS	14.6±0.1	17.7±0.3	15.6±0.2	23.8±0.2	44.3±0.2	50.4±0.3	49.0±0.1	53.6±0.2
	OFLS (40, 0.5)	14.6±0.1	17.7±0.2	15.7±0.1	23.9±0.2	44.3±0.3	50.4±0.3	49.0±0.2	53.6±0.3
OT w/Mean	SBS	14.6±0.1	17.8±0.3	15.6±0.2	23.9±0.1	44.5±0.1	50.5±0.3	49.0±0.2	53.6±0.2
	OFLS (40, 0.5)	14.6±0.1	17.8±0.2	15.6±0.2	23.9±0.1	44.5±0.1	50.5±0.3	49.0±0.2	53.6±0.2
BiMax w/Mean	SBS	14.7±0.1	17.6±0.1	15.9±0.1	24.0±0.2	44.4±0.1	50.5±0.1	49.3±0.1	53.8±0.1
	OFLS (40, 0.5)	14.7±0.1	17.8±0.1	15.9±0.1	24.0±0.3	44.5±0.2	50.5±0.1	49.3±0.3	53.9±0.4

Table 14: Auto-Rank: Docpair top 24,586, Sentpair top 400,000 Performance.

Document Alignment	Sentpair Num.	Bleu				ChrF			
		EMEA	EUB	EP	JRC	EMEA	EUB	EP	JRC
Baseline (Sloto et al., 2023)	2,654,090	18.1	20.1	18.4	25.7	49.4	53.0	52.1	55.7
Baseline (Github)	2,654,090	18.3	19.1	18.1	24.3	49.7	52.3	51.8	55.2
Baseline (Minh-Cong et al., 2023)	2,654,090	18.3	19.1	18.1	24.3	49.7	52.3	51.8	55.2
Baseline (Steingrimsson, 2023)	2,654,090	18.2	19.1	17.8	24.3	49.5	52.2	51.5	54.8
Baseline (Our)	2,654,090	18.3	19.1	18.1	24.4	49.6	52.3	51.8	55.2
Minh-Cong et al. (2023)	12,918,719	18.5	20.4	19.1	25.8	48.9	52.5	52.5	55.5
Steingrimsson (2023)	3,902,740	20.4	20.2	18.7	25.4	51.4	52.8	52.0	54.9
Margin Score 3.2M (Sloto et al., 2023)	3.2M	21.5	22.4	20.2	27.9	52.5	54.7	53.4	57.8
Mean-Pool (OFLS)$_{\|addbase}$	2,992,080	19.2	20.3	18.1	26.6	50.6	53.4	51.3	56.5
TK-PERT (OFLS)$_{\|addbase}$	2,997,590	19.2	20.4	18.0	26.4	50.5	53.1	51.1	56.1
OT w/Mean (OFLS)$_{\|addbase}$	2,985,377	19.2	20.2	18.7	26.5	50.5	53.2	52.1	56.4
BiMax w/Mean (OFLS)$_{\|addbase}$	2,958,214	19.2	20.4	18.3	26.7	50.4	53.3	51.7	56.4

Table 15: The results of "document alignment methods + baseline" compared to previous works.

tween the SBS and OFLS segmentation strategies, except for a slight advantage exhibited by BiMax over the other three methods in the EP and JRC-Acquis domains. This phenomenon may be attributed to multiple factors: the process from document alignment to final dataset construction involves numerous intermediate stages, and variability introduced at any of these stages may contribute to the observed homogenization of results.

G.2.4 Developing MT Benchmark compared to Previous Work

In addition to comparing the various methods among themselves, we also aim to compare our results against those of the WMT23 data task participants and organizers. However, the dataset we construct via document alignment and hierarchical mining can only serve as a high-quality but small-scale dataset, and we still lack a large-scale base dataset. Since we do not explore sentence filtering methods in depth, we utilize the organizers' baseline dataset (Sloto et al., 2023), which consists of sentence pairs obtained by taking the top-1 cosine similarity from the LASER embeddings, as our base dataset. We then augment it with the dataset we create. We prioritize our dataset by removing duplicates from the baseline dataset and including all sentence pairs derived from the 24,586 docpairs. Moreover, we rely exclusively on perplexity (PPL) to determine the best checkpoint.

As the results shown in Table 15, first, in comparison with the baseline method, we add less than one-fifth of its data size yet achieve a substantial improvement in accuracy in the EMEA, EUBookshop, and JRC-Acquis domains, indicating the high quality of our document-alignment-derived dataset. Second, compared with other participants, we achieve comprehensive high BLEU and ChrF in the JRC-Acquis domain compared to two participants, and also outperform Minh-Cong et al. (2023) in the EMEA domain. However, we observe that the organizers' baseline results on the EUbookshop and JRC-Acquis domains are substantially higher than both ours and those of other participants, likely due to differences in system environments or some other reasons. Accordingly, we refrain from direct comparison with their reported numbers (Sloto et al., 2023). Nonetheless, given that their pipeline employs margin scores for translation data mining—providing a clear performance advantage—we hypothesize that replacing our base dataset with one extracted using margin scores may further enhance our results.

G.3 Summarization of MT Benchmark for Document Alignment on the WMT23 Data Task

In Appendix G, we aim to develop an end-to-end system for the WMT23 data task, evaluating document alignment quality through its impact on downstream machine translation (MT) performance. We endeavor to present the development process with transparency and rigor. However, the final results exhibit a high degree of homogenization across methods (as shown in Table 14). As discussed in Appendix G.2.3, numerous intermediate variables exist in the process from document alignment to the construction of the final parallel sentence pair dataset. Additionally, factors such as the limited size of the Common Hostname data and the characteristics of the Estonian-Lithuanian language pair likely contribute to deviated results from our expectations. Furthermore, due to the absence of ground-truth document pairs in the WMT23 data task, the expected comparison of the four alignment methods is based on their performance on the MnRN dataset and the WMT16 test data; consequently, we cannot draw definitive conclusions about their relative effectiveness in the WMT23 data task.

Nonetheless, the results remain meaningful, as we carefully controlled all experimental variables. BiMax slightly outperforms OT while offering a noticeably faster processing speed (as shown in Table 13), indicating that parallel sentence pair datasets generated using BiMax can match OT in quality while requiring fewer computational resources. Moreover, as described in Section G.2.4, document alignment holds strong potential for producing high-quality translation data. Simply appending a basic baseline dataset can enable performance that rivals—or even exceeds—that of more complex data construction pipelines designed by WMT23 participants.

As mentioned in Section 1, the advancement of large language models (LLMs) has rendered document-level translation increasingly feasible. Therefore, rather than adhering to the conventional practice of evaluating alignment quality using downstream sentence-level MT systems, it may be more effective to assess document alignment directly through document-level machine translation.