

SNAPP: Segment-Level Neural Alignment Post-Processing for Low-Resource Parallel Sentence Mining

Niklas Hettich¹ and Shu Okabe^{1,2} and Alexander Fraser^{1,2,3}

¹School of Computation, Information and Technology, Technische Universität München (TUM)

²Munich Center for Machine Learning

³Munich Data Science Institute

Corresponding authors: {niklas.hettich, shu.okabe}@tum.de

Abstract

Machine Translation (MT) for low-resource languages has been challenging so far, mainly due to the small size or lack of parallel sentences. Parallel sentence mining and filtering are two automatic approaches which aim to identify and curate such resources. Yet, they usually rely on multilingual representation as a backend, which is known to be of poorer quality for low-resource language pairs, leading to the introduction of noisy pairs. We devise SNAPP, a post-processing pipeline which combines a segment-level filtering technique with an unsupervised neural aligner to remove highly similar but non-parallel sentence pairs. Modular components further enable more specific adaptation to the language pair under consideration. We evaluate filtering performance on three language pairs of varying typological distance. Finally, we show improvement on the downstream MT performance with our post-processing pipeline for the two harder language pairs.

1 Introduction

While Neural Machine Translation (NMT) achieves impressive performance in high-resource settings (Vaswani et al., 2017), its effectiveness drops significantly when applied to languages with limited parallel data (Kvapilíková and Bojar, 2023, *inter alia*). As the main bottleneck comes from the scarcity of data for such languages, a notable approach is to build such training corpora automatically from existing data. Parallel sentence mining, hence, aims to extract pairs of translated sentences from two monolingual corpora of comparable domains but not sentence-aligned (Zweigenbaum et al., 2017). However, embedding-based extraction approaches are prone to erroneously pairing sentences that exhibit high lexical similarity but diverge in meaning (Hangya and Fraser, 2019).

In this article, we hence introduce SNAPP (Segment-level Neural Alignment Post-

Processing), a post-processing pipeline designed to improve the quality of mined parallel corpora in low-resource settings. Our approach combines the segment-level filtering approach of Hangya and Fraser (2019) with the neural alignments from SimAlign (Jalili Sabet et al., 2020). We extend it with additional noise-reduction techniques, including vector space correction, stop-word filtering, and punctuation handling. We evaluate the effectiveness of our approach by applying the filtering pipeline to the BELOPSEM dataset (Okabe et al., 2025b), which features three low-resource language pairs with varying degrees of typological distance: Occitan–Spanish (OCI–ES), Chuvash–Russian (CHV–RU), and Upper Sorbian–German (HSB–DE). Our filtering pipeline yields performance improvements across all evaluated pairs, systematically increasing retrieval scores over the baseline mining pipeline.

Furthermore, we investigate the impact of the resulting corpus quality on downstream MT performance for the HSB–DE and CHV–RU language pairs. We compare the NLLB model (NLLB Team et al., 2024) across four distinct training configurations: a zero-shot baseline, alongside three fine-tuned variants based on mined and filtered translation pairs. We observe improvement thanks to our approach, as more parallel pairs are found.

Our main contributions are as follows: (1) We propose a post-processing pipeline for parallel sentence filtering that integrates an unsupervised neural aligner into a segment-level filtering approach, with modular components for language-specific adaptation. (2) We empirically demonstrate the robustness and effectiveness of our pipeline across three low-resource language pairs of varying typological distances. (3) We show that the improved parallel corpus quality directly leads to enhanced downstream NMT performance.

We publicly release the SNAPP pipeline.¹

2 Related Work

2.1 Parallel sentence mining and filtering

Mining Given the scarcity of parallel training data for NMT models in low-resource scenarios, parallel sentence mining aims to identify translation pairs in comparable monolingual texts, where sentences are not aligned. The BUCC 2017 Shared Task (Zweigenbaum et al., 2017) established a benchmark and evaluation protocol for the task, focusing on four high-resource language pairs. Later approaches improved mining quality with sentence-level embeddings to identify such translation pairs (i.a., Artetxe and Schwenk, 2019).

Filtering Mined corpora are known to contain non-translation pairs, which affect the quality of the downstream MT performance. Parallel sentence filtering aims to remove such noise to compile a high-quality parallel training data. The WMT 2018 and 2019 Shared Tasks (Koehn et al., 2018, 2019) focused on cleaning the provided noisy parallel corpora and evaluated their success based on the performance of the trained MT systems. While Koehn et al. (2018) provided noisy data for the high-resource English–German language pair, Koehn et al. (2019) introduced a low-resource scenario to increase the task’s complexity, where direct filtering via multilingual sentence embeddings proved highly effective (Chaudhary et al., 2019).

Building upon these efforts, Sloto et al. (2023) focused on the entire data curation process by providing participants with unaligned monolingual corpora before any sentence mining. In these sentence extraction pipelines, cross-lingual sentence embeddings have proven highly advantageous for initial candidate retrieval in low-resource scenarios (Chaudhary et al., 2019; Koehn et al., 2019). However, such embedding-based approaches are prone to erroneously pairing sentences that exhibit high lexical similarity but diverge in meaning (Hangya and Fraser, 2019). Therefore, recent Shared Task findings advocate for the subsequent application of robust alignment and data filtering steps (Sloto et al., 2023), for instance, through a combination of fine-grained sentence alignment and multi-stage filtering heuristics (Steingrimsson, 2023).

Another approach to remove noisy pairs from parallel corpora is to rely on classification tools,

such as Bicleaner (Zaragoza-Bernabeu et al., 2022) or the approach of Jiang et al. (2026). They need, however, to be trained on parallel corpora. We do not consider classification systems, as such translation pairs are scarce.

2.2 Mining for low-resource languages

Multilingual embedding-based methods have also been successfully applied to mine parallel data to noticeably improve low-resource Machine Translation (Kvapilíková and Bojar, 2023). Yet, previous work mostly relied on fine-tuning or distilling existing multilingual models with parallel data (Hefernan et al., 2022; Tan et al., 2023), while such resources are limited for low-resource language pairs. The PaSeMiLL mining pipeline (Okabe and Fraser, 2025) provides a standardized setting to evaluate such embedding-based extraction methods in low-resource settings. The pipeline computes sentence embeddings by mean-pooling token embeddings extracted from a pre-trained language model. To identify semantically related sentence pairs, it evaluates candidate similarities using the Cross-Domain Similarity Local Scaling (CSLS) metric (Lample et al., 2018), which mitigates the hubness problem in vector spaces. Subsequently, candidate pairs are filtered out if their similarity score falls below a threshold computed dynamically (Okabe and Fraser, 2025). Additionally, the implementation allows for the application of Cluster-based Isotropy Enhancement (CBIE) (Rajae and Pilehvar, 2021), which locally corrects embedding biases to improve overall vector-space isotropy before the extraction.

3 Background

3.1 UnsupPSE: Segment-level filtering

To filter out sentence pairs with high lexical overlap but diverging meaning, Hangya and Fraser (2019) propose ‘UnsupPSE’, an algorithmic segment-level filtering approach. Operating strictly on the word level, the method greedily aligns each source word s_i to its most similar target word t_j using static embeddings evaluated via the CSLS metric and orthographic similarity. Unaligned tokens receive a similarity value of zero. To accommodate a small amount of unaligned words, similarity scores are smoothed using a window-based local average. A token sequence forms a contiguous segment if all averaged scores exceed a predefined threshold. Figure 2 in Appendix A shows the window-based

¹<https://github.com/TUM-NLP/SNAPP>

averaging of a HSB–DE candidate pair.

Each segment is matched at most once to the counterpart sharing the highest number of word alignments. To reduce noise, candidate segment pairs are discarded if they fail to meet constraints on minimum length or maximum length difference (default: 5 words).

Finally, the overall similarity of the candidate pair is evaluated using a combined metric that weights the sum of all alignment similarities by the length of the longest segment, normalized by the respective lengths of the source and target sentences. The candidate pair is retained if this final score exceeds a tunable threshold.

3.2 SimAlign: Neural word alignment

Statistical alignment methods, such as `fast_align` (Dyer et al., 2013), require large amounts of parallel data for reliable bilingual alignments, which makes them ill-suited to low-resource scenarios. This motivates the use of unsupervised neural aligners, which rely primarily on semantic similarity rather than co-occurrence. A state-of-the-art neural alignment tool is SimAlign (Jalili Sabet et al., 2020), which leverages pre-trained multilingual word embeddings to determine the semantic similarity between source and target tokens. This neural unsupervised approach is particularly beneficial in low-resource scenarios, as it allows for zero-shot use without requiring parallel training data. By relying on cross-lingual transfer, it can establish reliable mappings provided that the underlying multilingual language model offers meaningful semantic representations for the specific low-resource language. SimAlign provides three distinct alignment heuristics (Argmax, Itermax, and Match), which offer different trade-offs between alignment precision and recall. We detail the methods in Appendix B.

4 Filtering pipeline

Our SNAPP filtering pipeline essentially combines the UnsupPSE approach (Hangya and Fraser, 2019) with the neural alignment tool SimAlign (Jalili Sabet et al., 2020). Figure 1 provides a summarized overview of the pipeline.

4.1 Pre-processing

Tokenization and embedding The filtering pipeline receives a candidate sentence pair in the source and target languages. Using a multilingual pretrained language model (mPLM) and its tokenizer, the pipeline tokenizes both sentences and

extracts their contextualized embeddings. Furthermore, as in the UnsupPSE approach, subword tokens are aggregated to the word level (based on whitespace and punctuation boundaries) by averaging their respective embeddings. An example of this token aggregation is provided in Appendix C.

Mean vector subtraction To mitigate the ‘Cone Effect’ (Ethayarajh, 2019), the resulting vectors are then centered by subtracting the language-specific mean vector. This mean vector is computed before the pipeline’s execution by averaging the embeddings of all tokens from the initial monolingual corpus of the respective language. We consider two computation methods: either from a fixed monolingual corpus only (‘static’) or from the dataset to process (‘dynamic’) for better adaptability.

4.2 Word alignment and token similarity

Word alignment After the pre-processing step, we apply the neural alignment tool SimAlign (Jalili Sabet et al., 2020) to words. Following Jalili Sabet et al. (2020), we extract contextualized representations from the 8th layer of the language model across the entire pipeline, including for SimAlign. Among its three alignment heuristics, we mainly use the Argmax strategy due to its stricter alignment criterion (cf. Appendix B). It notably leads to high precision with minimal false positives that could negatively affect later scoring. These word alignments are later used to compute cosine similarity scores instead of CSLS. Because our pipeline processes candidate pairs sequentially, computing the neighborhood-based CSLS score for contextualized word representations across the entire vocabulary would be computationally prohibitive. Instead, pairing standard cosine similarity with mean-vector subtraction provides an efficient and effective alternative to counteract vector space anisotropy. We rely on the aggregated word-level embeddings to assess the semantic similarity of the aligned pairs.

Stop-word filtering To further reduce noise, the filtering pipeline can additionally ignore *unaligned* stop-words given their limited semantic value. Due to typological differences in a language pair, such words can be less frequent or absent in one of the two languages (e.g., articles in Slavic languages). Since they can break segment-level alignments on one or both sides of a pair, we allow this step to be applied independently to the source (‘src’) or the target (‘trg’). This post-processing module relies

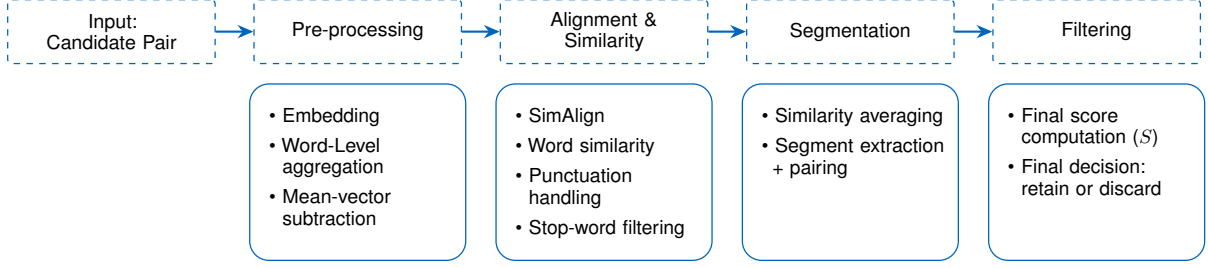


Figure 1: Simplified visualization of the proposed filtering approach: SNAPP.

on a list of stop-words. Appendix D describes how we collected the stop-words for three of the languages which did not have available lists.

4.3 Parallel segment extraction

Following the segment-level approach of Hangya and Fraser (2019), the cosine similarity scores are averaged using a window-based method, to bridge small semantic gaps between token alignments (cf. Figure 2). Crucially, this smoothing compensates for certain missed alignments (i.e., false negatives) from the previous step. However, it is simultaneously highly sensitive to false-positive alignments, which would artificially inflate adjacent scores during the averaging process. Thus, this windowing technique complements the conservative Argmax alignment strategy well. Tokens with an averaged similarity score exceeding a tunable threshold are aggregated into contiguous segments. The target segments are then mapped to their respective source segments via a greedy assignment strategy based on the number of shared alignments. We enforce a 0.25 upper bound on the relative length difference between paired segments. This step aims to identify truly parallel segments while simultaneously mitigating additional noise by penalizing unaligned or non-parallel text spans.

4.4 Filtering

Finally, we compute a filtering score S according to Equation (1) to determine whether a candidate pair should be retained. This formula formalizes the segment-weighted scoring metric as implemented in Hangya and Fraser (2019):

$$S(s, t) = \frac{\sum_{(s_i, t_j) \in A} \text{sim}(s_i, t_j)}{|t|} \cdot \frac{|seg_{max}|}{|s|}, \quad (1)$$

where A represents the set of all aligned word pairs (s_i, t_j) obtained via SimAlign (Jalili Sabet et al., 2020), $\text{sim}(s_i, t_j)$ denotes their cosine similarity score, seg_{max} corresponds to the longest extracted

segment in the source sentence s , and $|s|$ and $|t|$ represent the respective lengths of the source and target sentences after any applied stop-word removal.

Although the formula remains identical to that of Hangya and Fraser (2019), the underlying components are derived through distinct methodologies. Specifically, we substitute the static dictionary-based word alignments with SimAlign, replace the CSLS score with the cosine similarity, and we additionally remove stop-words, notably altering the algorithm’s behavior. If the score S exceeds a tunable threshold α , the candidate sentence pair is retained, otherwise it is discarded.

5 Sentence filtering experiments

5.1 Sentence mining setup

5.1.1 Datasets and languages

We evaluate the efficacy of our SNAPP filtering approach and focus on the language pairs in BELOPSEM,² a mining benchmark for low-resource languages (Okabe et al., 2025b). It features three language pairs of varying language proximity (and hence difficulty): Occitan–Spanish, Upper Sorbian–German, and Chuvash–Russian. Here, each low-resource source language is paired with a high-resource target language.

The design of the corpus follows the established BUCC Shared Task method (Zweigenbaum et al., 2017): true translation pairs have been randomly injected in two comparable but non-parallel monolingual corpora. The aim is hence to retrieve only the parallel sentences with minimal noise.

Table 1 details the sentence counts for each dataset of a language pair in BELOPSEM. The training set is a small set to set the filtering threshold, while the test set is larger to replicate an actual mining scenario.

²<https://github.com/shuokabe/Belopsem>

Language pairs	Train			Test		
	source	target	parallel	source	target	parallel
OCI-ES	7,899	7,780	486	23,675	23,334	1,457
CHV-RU	7,998	7,994	499	23,995	23,985	1,498
HSB-DE	15,980	15,999	1,000	47,847	47,999	3,000

Table 1: Number of sentences in the BELOPSEM dataset.

5.1.2 Baseline mining systems

To establish a robust baseline for evaluating our approach, we employ the PaSeMiLL mining pipeline. We consider two settings where Glot500-m (Imani et al., 2023) serves as the model backend, following (Okabe et al., 2025b). This mPLM extends XLM-R (Conneau et al., 2020) specifically to low-resource languages. It notably covers the three low-resource languages of the mining benchmark.

First, we consider its setting with the improved cross-lingual representation (with Cluster-Based Isotropy Enhancement, or CBIE) and denote it as the ‘baseline’. We also report the performance with the pipeline’s native post-processing approach as ‘baseline+PP’, where sentences are ranked according to the ratio of aligned words in a sentence pair. This setting also uses a lower similarity threshold to start from a larger pool of sentence pairs (and thus a higher recall value).

5.1.3 Mining evaluation

We assess the quality of the final mined and filtered corpora using the standard precision (‘P’), recall (‘R’), and F1-score (‘F’) metrics with the provided gold sentence pairs. We also report the number of sentence pairs (‘#Pairs’) after mining or filtering.

5.2 Post-processing: Main results

Overall performance Table 2 illustrates the performance gains achieved by our post-processing approach in its optimal configuration for each language pair. Compared to both the standard baseline and its native post-processing setting, our filtering pipeline consistently increases the F1-score across all three language pairs. Applying this post-processing step allows the PaSeMiLL mining pipeline to operate with a low filtering threshold (here: -1.0) in order to maximize initial recall. Although this approach successfully increases recall, it inherently leads to a substantial rise in false positives. While our filtering pipeline yields precision for OCI-ES that is comparable to the base PaSeMiLL mining pipeline, it demonstrates a notable improvement in precision for CHV-RU

Method	P	R	F	ΔF	#Pairs
Occitan-Spanish (OCI-ES)					
baseline	90.50	77.14	83.29	Ref.	-
baseline+PP	90.38	79.27	84.46	+1.17	-
PaSeMiLL (thr. = -1.0)	5.42	88.06	10.21	-	23,675
SNAPP (fast_align)	94.06	70.69	80.72	-2.57	1,095
SNAPP (best)	89.62	81.74	85.50	+2.21	1,329
Chuvash-Russian (CHV-RU)					
baseline	46.67	37.38	41.51	Ref.	-
baseline+PP	54.62	36.32	43.62	+2.11	-
PaSeMiLL (thr. = -1.0)	3.62	57.94	6.81	-	23,995
SNAPP (fast_align)	67.85	20.43	31.40	-10.10	451
SNAPP (best)	55.83	44.73	49.67	+8.16	1,200
Upper Sorbian-German (HSB-DE)					
baseline	60.74	33.47	43.15	Ref.	-
baseline+PP	61.03	43.53	50.82	+7.67	-
PaSeMiLL (thr. = -1.0)	3.48	55.50	6.55	-	47,847
SNAPP (fast_align)	89.73	30.00	44.97	+1.82	1,003
SNAPP (best)	80.26	49.73	61.41	+18.26	1,859

Table 2: Post-processing performance with Precision (P), Recall (R), and F1-score (F) across the three language pairs. ‘ ΔF ’ presents the F1-score gain against the baseline. Baseline results are referenced from Okabe et al. (2025b) with the CBIE configuration.

and HSB-DE. Consequently, the combination of higher recall and a comparable or improved precision yields more accurate overall results than the baseline PaSeMiLL mining pipeline. The hyperparameters used to obtain the results are presented in Appendix E.

Regarding computational cost, our pipeline processes approximately 5,000 candidate pairs per hour on a single NVIDIA A100 GPU (80GB VRAM).

Neural vs. statistical alignment Furthermore, Table 2 compares the performance of our filtering approach when using neural versus statistical alignments. Specifically, we evaluate the neural aligner SimAlign (Jalili Sabet et al., 2020) against a baseline employing the fast_align statistical tool (Dyer et al., 2013). Both pipeline variants were executed using identical hyperparameter configurations. Fast_align requires the entire corpus to generate alignments, whereas our pipeline processes candidate pairs sequentially. To accommodate this, we pre-computed and filtered the statistical alignments for the PaSeMiLL candidate pairs and accessed them dynamically during our post-processing phase. While statistical tools rely on large parallel corpora and often struggle under low-resource or noisy conditions, neural aligners leverage pre-trained language models to operate effec-

tively in zero-shot scenarios. Consequently, SimAlign yields a substantial performance improvement over fast_align as shown in Table 2. Appendix F further details the underlying causes of imprecise alignments produced by fast_align in our scenario and discusses various settings for filtering noise before running the pipeline.

5.3 Ablation studies

Our pipeline allows for the activation and configuration of various components to adapt filtering to the characteristics of the language pair. This section evaluates the impact of these diverse configurations on the overall filtering performance across the BELOPSEM language pairs. As the punctuation handling configuration did not yield notable performance improvements, a detailed analysis of this component is deferred to Appendix G.

We evaluate three components of our pipeline: (1) The mean-vector subtraction component (‘mean vect.’) applied to embeddings in three configurations: no subtraction (‘w/o’), static mean vector (‘s’; using the training set), or dynamic mean vector (‘d’; using the training or test set). (2) The stop-word filtering component (‘SW’), which can be applied to the source (‘src’) or target (‘trg’) languages. (3) The SimAlign alignment method, where words can be aligned according to the three heuristics: Argmax (‘A’), Itermax (‘I’), or Match (‘M’). Tables 3, 4, 5, and 10 evaluate the impact of individual components by varying one factor at a time while keeping all other parameters fixed at the optimal configuration for the respective language pair.

We stress here that the optimal configurations were strictly selected based on their performance on their respective *training* sets. The tables report the results of these selected configurations evaluated on the unseen test sets, with the chosen optimal configuration for each language pair highlighted in gray. Notably, punctuation handling was enabled exclusively for OCI-ES and remained disabled for both CHV-RU and HSB-DE. We perform our ablation study starting from the optimal setting (which depends on the language pair), varying one factor at a time.

5.3.1 Mean vector subtraction

Table 3 displays the effect of mean vector subtraction on the word embeddings during the word-level aggregation stage, as described in Section 4.1. We observe that applying this module significantly im-

Lang Pair	mean vect.	SW src	SW trg	Sim-Align	P	R	F	# Pairs
OCI-ES	w/o	No	No	A	34.53	84.56	49.03	3,568
	s	No	No	A	89.55	81.74	85.47	1,330
	d	No	No	A	89.62	81.74	85.50	1,329
CHV-RU	w/o	Yes	Yes	I	8.15	51.74	14.07	9,515
	s	Yes	Yes	I	54.65	44.73	49.19	1,226
	d	Yes	Yes	I	55.83	44.73	49.67	1,200
HSB-DE	w/o	No	Yes	A	10.10	54.40	17.03	16,162
	s	No	Yes	A	80.26	49.73	61.41	1,859
	d	No	Yes	A	80.25	49.70	61.38	1,858

Table 3: Ablation study: Mean vector configuration. We keep the other settings fixed compared to the best setting (notably Itermax for CHV-RU).

proves performance by helping cross-lingual alignment. Without mean vector subtraction, the post-processing stage retains most of the false positives extracted by the initial PaSeMiLL mining pipeline. The severe drop in precision (compared to the best-performing configuration) and the high recall indicate the negative effect of anisotropy in the raw vectors provided by Glot500-m. Without centering the vector space, the embeddings tend to point in a similar direction, leading to high cosine similarity scores even for wrongly aligned tokens (Ethayarajh, 2019). Thus, the subsequent window-based averaging yields inflated final filtering scores and allows numerous false positives to remain after the final filtering step. By subtracting the mean vector for each language, the respective embedding spaces are shifted towards the origin, effectively mitigating this directional imbalance.

Regarding the computation method, we observed little difference between both static and dynamic mean vectors; hence, applying this step is crucial, regardless of the method. Ultimately, this subtraction enables a more robust distinction between true translations and noise, thereby yielding higher-quality filtering during post-processing.

5.3.2 Alignment methods

Table 4 displays the effect of the three alignment methods in SimAlign. We observe that Argmax yields the best overall performance for the OCI-ES and HSB-DE datasets. This method benefits from a conservative selection strategy that retains only mutually maximal alignments, thereby increasing the probability of correctness. The subsequent window-based averaging effectively compensates for the omitted alignments, resulting in high precision while maintaining a high recall.

Lang Pair	mean vect.	SW src	SW trg	Sim-Align	P	R	F	# Pairs
OCI-ES	d	No	No	A	89.62	81.74	85.50	1,329
	d	No	No	I	85.57	83.05	84.29	1,414
	d	No	No	M	83.92	83.12	83.52	1,443
CHV-RU	d	Yes	Yes	A	60.26	39.79	47.93	989
	d	Yes	Yes	I	55.83	44.73	49.67	1,200
	d	Yes	Yes	M	56.18	43.39	48.96	1,157
HSB-DE	s	No	Yes	A	80.26	49.73	61.41	1,859
	s	No	Yes	I	77.24	48.30	59.43	1,876
	s	No	Yes	M	78.33	41.70	54.43	1,597

Table 4: Ablation study: SimAlign alignment method.

In contrast, we note a performance drop for Argmax on the CHV-RU dataset compared to Itermax and Match. Belonging to distinct language families (Turkic and Slavic, respectively), the CHV-RU pair exhibits the largest typological distance among the evaluated datasets, with fewer direct one-to-one word correspondences. Consequently, the strict alignment criteria of Argmax produce too few alignments compared to Itermax and Match. This severe sparsity cannot be fully mitigated by the window-based averaging, causing true positives to fall below the filtering threshold.

Since Itermax and Match inherently lead to a greater number of alignment links, they prove advantageous for this language pair with a more substantial typological distance. While both algorithms inevitably introduce more false alignments (lowering precision), the substantial gain in recovered true positives (higher recall) effectively offsets this drawback, yielding a higher overall F1-score.

5.3.3 Stop-word filtering

We investigate the impact of filtering unaligned stop-words in Table 5. In general, stop-word filtering bridges minor semantic gaps, yielding larger contiguous segments and a subsequent increase in overall recall (e.g., by up to 12.90 percentage points for HSB-DE). Simultaneously, however, the volume of noise within the remaining pairs increases, which leads to a decline in precision (dropping by 12.87 percentage points for HSB-DE). It emerges that the impact of stop-word filtering depends on the linguistic distance between the source and target languages. While the pipeline does not benefit from this filtering for OCI-ES, it contributes to the performance for CHV-RU and substantially for HSB-DE. The latter two language pairs exhibit a larger linguistic distance compared to the closely related OCI-ES.

Lang Pair	mean vect.	SW src	SW trg	Sim-Align	P	R	F	# Pairs
OCI-ES	d	No	No	A	89.62	81.74	85.50	1,329
	d	No	Yes	A	84.36	82.16	83.24	1,419
	d	Yes	No	A	85.82	82.22	83.98	1,396
	d	Yes	Yes	A	79.16	85.24	82.09	1,569
CHV-RU	d	No	No	I	60.95	40.32	48.53	991
	d	No	Yes	I	55.88	44.06	49.27	1,181
	d	Yes	No	I	60.90	40.45	48.62	995
	d	Yes	Yes	I	55.83	44.73	49.67	1,200
HSB-DE	s	No	No	A	88.18	38.53	53.63	1,311
	s	No	Yes	A	80.26	49.73	61.41	1,859
	s	Yes	No	A	86.29	39.03	53.75	1,357
	s	Yes	Yes	A	75.31	51.43	61.12	2,049

Table 5: Ablation study: Stop-word filtering.

6 Downstream evaluation: MT

6.1 Methodology

To evaluate the impact of the proposed filtering on the quality of the training corpus for MT systems, we compare the performance of NLLB (facebook/nllb-200-distilled-600M) (NLLB Team et al., 2024) in different settings for the two harder language pairs: HSB-DE and CHV-RU.

6.1.1 Datasets

Training To replicate an actual parallel data curation scenario, we rely on the correspondence between Wikipedia articles, in which some sentences are naturally parallel. We use the Wikipedia dataset provided by Wikimedia³ as our initial set of ‘comparable’ monolingual corpora. We only consider articles whose titles match; the resulting datasets mainly contain named entities (mostly place and person names). Moreover, since articles are usually longer on the (target) high-resource side than the (source) low-resource language, we cap their length to 1.5 times the number of sentences of the low-resource source side.

Evaluation We evaluated the performance of the NLLB model for HSB-DE on the development set of the WMT 2025 Shared Task on LLMs with Limited Resources for Slavic Languages (Okabe et al., 2025a).⁴ For CHV-RU, we use the established FLORES+ MT benchmark (NLLB Team et al., 2024) which extends the previous version from Guzmán et al. (2019) and Goyal et al. (2022).

³<https://huggingface.co/datasets/wikimedia/wikipedia>

⁴<https://github.com/TUM-NLP/llms-limited-resources2025/tree/main/Sorbian/hsb/MT>

	HSB-DE			CHV-RU		
	HSB	DE	source	CHV	RU	source
train	34,445	42,668	Wikipedia	111,156	139,327	Wikipedia
test		4,000	WMT		1,012	FLORES+

Table 6: Number of sentences in the monolingual training set before mining and in the MT test set.

Table 6 summarizes the dataset sizes before the mining step (i.e., after the title matching and truncation of the target side) and the evaluation set.

6.1.2 MT methodology

We mine and filter the datasets from Section 6.1.1 and then fine-tune the NLLB model (NLLB Team et al., 2024) under four configurations.

Baseline We consider NLLB in a zero-shot setting as a baseline. While German and Russian were included in the pre-training corpus of the model, Upper Sorbian and Chuvash were not (NLLB Team et al., 2024). Consequently, zero-shot evaluation is only applicable in the low-resource to high-resource translation direction. For this setting, we feed the input text into NLLB using its closest supported language proxy as the source language code: Czech (ces_Latn) for Upper Sorbian and Kazakh (kaz_Cyrl) for Chuvash.

Fine-tuning configurations Beyond the zero-shot setting, we consider three fine-tuning configurations. First, we use the corpus mined with the base PaSeMiLL pipeline (‘baseline’ in Table 2) configured with the optimal dynamic filtering threshold for each language pair ($\lambda = 1.5$ for HSB-DE and $\lambda = 1.8$ for CHV-RU). We denote this setting ‘FT-mined’. Both subsequent configurations are based on the candidate pairs, retrieved by the PaSeMiLL mining pipeline with a static filtering threshold of $\lambda = -1.0$ to maximize initial recall. We then apply our post-processing pipeline configured with the best-performing hyperparameters (Table 8). In the third configuration, we assess our best post-processing pipeline to curate sentence pairs (‘FT-sentences’). Finally, we also extract the longest contiguous segment pairs (‘FT-segments’) for each candidate pair. This setting aims to keep parallel *segments* from non-parallel (but similar) sentence pairs, including those below the final filtering threshold.

MT training Following the established paradigm of rapid adaptation for low-resource languages (Neubig and Hu, 2018), we fine-tune the

facebook/nllb-200-distilled-600M architecture across the three evaluated settings. For this, we follow Ahmadi et al. (2025) and expanded the tokenizer by introducing ‘hsb_Latn’ and ‘chv_Cyrl’ as new language tokens. We initialized their embeddings with the pre-trained weights from closely related, better-resourced anchor languages. Specifically, we use Czech (‘ces_Latn’) for Upper Sorbian, and Kazakh (‘kaz_Cyrl’) for Chuvash, as both belong to the same language family and share the same script.

All configurations were fine-tuned for 3 epochs using the AdamW optimizer with an initial learning rate of 2×10^{-5} , a weight decay of 0.01, a per-device batch size of 8, and a maximum sequence length of 128 tokens.

6.1.3 Evaluation

For each language pair, we translate sentences in both directions. We evaluate the quality of the translation using the standard BLEU (Papineni et al., 2002) and chrF2 scores (Popović, 2015), computed with the sacreBLEU toolkit (Post, 2018).⁵ We do not consider neural metrics to keep a consistent evaluation setting across all directions; Upper Sorbian and Chuvash are currently not supported by such systems.

6.2 MT results

Table 7 displays the MT performance with NLLB for both directions of the two language pairs. We observe that in all four translation directions, our filtering pipeline leads to improved MT scores compared to the zero-shot setting, when applicable, and the standard mining system (FT-mined), based on the chrF2 score. Between the sentence and segment extraction, we do not see a clear trend. Since the larger size of the training corpus does not imply a performance increase, the extracted segments appear to contain more samples but more noise. To verify translation performance gains, we conducted paired bootstrap resampling (1,000 resamples) using sacreBLEU (Post, 2018). Improvements over the baseline are statistically significant (p -value < 0.01), with full p -values reported in Appendix H.

To ensure that our performance gains are driven by data quality rather than simply increased data

⁵sacreBLEU signatures: BLEU: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.5.1; chrF2: nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.5.1.

Direction	Setting	# Pairs	# Words	chrF2	BLEU
HSB → DE	zero-shot	-	-	35.60	9.20
	FT-mined	2,169	32,647	42.60	15.30
	FT-sentences	3,919	50,972	45.10 [*] †	17.50 [*] †
	FT-segments	11,622	113,272	43.20 [*] †	14.40 [*] †
DE → HSB	FT-mined	2,169	32,647	27.20	4.20
	FT-sentences	3,919	50,972	28.90 [*] †	5.00 [*] †
	FT-segments	11,622	113,272	29.60 [*] †	5.60 [*] †
CHV → RU	zero-shot	-	-	21.00	1.60
	FT-mined	7,789	55,866	21.00	2.50
	FT-sentences	26,877	249,881	31.60 [*] †	7.10 [*]
	FT-segments	69,568	735,944	33.10 [*] †	7.40 [*]
RU → CHV	FT-mined	7,789	55,866	20.00	1.40
	FT-sentences	26,877	249,881	23.70 [*] †	2.50 [*]
	FT-segments	69,568	735,944	22.80 [*] †	2.30 [*]

Table 7: Downstream machine translation performance of the NLLB model for HSB–DE and CHV–RU. ‘# Pairs’ indicates the number of sentence or segment pairs used for fine-tuning. ‘# Words’ represents the number of words in the high-resource language. Superscript * denotes statistically significant improvement (or difference) over ‘FT-mined’, while † indicates a significant difference between ‘FT-sentences’ and ‘FT-segments’ (p -value < 0.01).

volume, we additionally evaluated the models using fine-tuning corpora with a matching number of words on the high-resource language side. Results and analysis for this experiment are detailed in Appendix I. Under this matched setting, the FT-sentences configuration still leads to better translation quality than FT-mined, whereas FT-segments can fall below it.

7 Conclusion

We present SNAPP, a parallel sentence filtering system, which post-processes sentence pairs based on unsupervised neural alignments to identify their matching *segments*. SNAPP aims to address one weakness of mean-pooled embedding approaches in mining, where sentence pairs with high lexical overlap but different meanings could not be discarded. Our system was mainly applied to low-resource pairs, which suffer more from the issue due to poorer multilingual representation. Our pipeline yields improved mining quality for three low-resource language pairs compared to the existing baseline, PaSeMiLL. We further included components for language-specific adjustments; word-level mean vector subtraction led to notably better filtering performance thanks to fewer false positives. Further improvements were also observed from stop-word filtering and the alignment strategy in SimAlign, depending on the language distance between the source and target languages. Finally,

we examined the impact of this improved corpus quality on the downstream performance of machine translation systems. By evaluating the NLLB model (NLLB Team et al., 2024) across four distinct configurations (a zero-shot baseline, alongside fine-tuning on mined PaSeMiLL sentences, post-filtered sentences, and post-filtered segments), we observed improved translation performance thanks to our proposed filtering system compared to the base mining pipeline.

Future work will extend the approach to other low-resource language pairs, specifically with larger typological differences (as in CHV–RU). We will further investigate the conditions in which extracted *segments* can consistently improve MT performance and outperform mined sentence pairs.

Limitations

First, the performance of our proposed filtering approach is inherently bounded by the recall achieved during the initial sentence-mining phase (in our case, the PaSeMiLL mining pipeline; Okabe and Fraser, 2025). All sentence pairs missed at this step will be irrecoverable by our system.

A major factor influencing performance for both mining and filtering in our experiments is the embedding quality produced by the underlying language model. Glot500-m was pre-trained on monolingual sentences for all three low-resource languages. Our approach may, hence, yield different results for other languages and models, depending on their pre-training data.

We mainly focus on Glot500-m and do not consider more advanced sentence embedding approaches, such as LaBSE (Feng et al., 2022) or SONAR (Duquenne et al., 2023). As our goal is to mine and filter parallel sentences for low-resource languages, we posit that such resources are not available to fine-tune (Kvapilíková and Bojar, 2023) or distill (Heffernan et al., 2022; Tan et al., 2023) the sentence embedding models. The mining and filtering approach in our work only relies on monolingual pre-training (Glot500-m), which is more accessible to low-resource languages.

Moreover, the compiled stop-word lists can be considered another limiting factor due to their heuristic construction. While our pipeline operates unsupervised, relying only on monolingual pre-training, using stop-word filtering introduces a reliance on human effort and external resources.

Specifically, manually verifying the 300 most frequent candidate words required approximately two hours per language and depended on the availability of pre-annotated corpora (e.g., UD treebanks) or existing translation tools. For extremely low-resource languages, such datasets and systems are often not available, thereby restricting the portability of this specific component. Solely relying on function words for the creation of the stop-word list does not cover all language-specific stop-words.

Acknowledgments

We thank the anonymous reviewers for their comments. This work was partly funded by the European Union (ERC, EPICAL, 101141712). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

The authors gratefully acknowledge the scientific support and resources of the AI service infrastructure *LRZ AI Systems* provided by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW), funded by Bayerisches Staatsministerium für Wissenschaft und Kunst (StMWK).

References

- Sina Ahmadi, Rico Sennrich, Erfan Karami, Ako Marani, Parviz Fekrazad, Gholamreza Akbarzadeh Baghban, Hanah Hadi, Semko Heidari, Mahir Dogan, Pedram Asadi, Dashne Bashir, Mohammad Amin Ghodrati, Kourosh Amini, Zeynab Ashourinezhad, Mana Baladi, Farshid Ezzati, Alireza Ghasemifar, Daryoush Hosseinpour, Behrooz Abbaszadeh, and 14 others. 2025. [PARME: Parallel corpora for low-resourced Middle Eastern languages](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30032–30053, Vienna, Austria. Association for Computational Linguistics.
- Mikel Artetxe and Holger Schwenk. 2019. [Margin-based parallel corpus mining with multilingual sentence embeddings](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.
- Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O’Reilly Media, Inc.
- Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. [Low-resource corpus filtering using multilingual sentence embeddings](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Lee R. Dice. 1945. [Measures of the amount of ecologic association between species](#). *Ecology*, 26(3):297–302.
- Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. [Sonar: Sentence-level multimodal and language-agnostic representations](#). *Preprint*, arXiv:2308.11466.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. [A simple, fast, and effective reparameterization of IBM model 2](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). 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 55–65, Hong Kong, China. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Ariavzhagan, 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.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 evaluation benchmark for low-resource and multilingual machine translation](#). *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc’Aurelio Ranzato. 2019. [The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English](#). 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 6098–6111, Hong Kong, China. Association for Computational Linguistics.
- Viktor Hangya and Alexander Fraser. 2019. [Unsupervised parallel sentence extraction with parallel segment detection helps machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1224–1234, Florence, Italy. 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.
- Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. [Glot500: Scaling multilingual corpora and language models to 500 languages](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.
- Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. [SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.
- Ruiyang Jiang, Shu Okabe, and Alexander Fraser. 2026. [Parallel sentence filtering for low-resource language pairs: A case study for Upper Sorbian, German, and Czech](#). In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 8883–8896, Palma de Mallorca, Spain. ELRA Language Resource Association.
- Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. [Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.
- Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. [Findings of the WMT 2018 shared task on parallel corpus filtering](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.
- Ivana Kvapilíková and Ondřej Bojar. 2023. [Boosting unsupervised machine translation with pseudo-parallel data](#). In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 135–147, Macau SAR, China. Asia-Pacific Association for Machine Translation.
- Guillaume Lample, Alexis Conneau, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. [Word translation without parallel data](#). In *International Conference on Learning Representations*.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- Graham Neubig and Junjie Hu. 2018. [Rapid adaptation of neural machine translation to new languages](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal Dependencies v2: An evergrowing multilingual treebank collection](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. [Scaling neural machine translation to 200 languages](#). *Nature*, 630(8018):841–846.
- Shu Okabe, Daryna Dementieva, Marion Di Marco, Lukas Edman, Katharina Haemmerl, Marko Měškank, Anita Hendrichowa, and Alexander Fraser. 2025a. [Findings of the WMT 2025 shared task LLMs with limited resources for Slavic languages: MT and QA](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 503–519, Suzhou, China. Association for Computational Linguistics.
- Shu Okabe and Alexander Fraser. 2025. [Bilingual sentence mining for low-resource languages: a case study on Upper and Lower Sorbian](#). In *Proceedings of the Eight Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 11–19, Honolulu, Hawaii, USA. Association for Computational Linguistics.
- Shu Okabe, Katharina Hämmel, and Alexander Fraser. 2025b. [Improving parallel sentence mining for low-resource and endangered languages](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–205, Vienna, Austria. 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.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Sara Rajae and Mohammad Taher Pilehvar. 2021. [A cluster-based approach for improving isotropy in contextual embedding space](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 575–584, Online. 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.
- Steinthor Steingrímsson. 2023. [A sentence alignment approach to document alignment and multi-faceted filtering for curating parallel sentence pairs from web-crawled data](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 366–374, Singapore. Association for Computational Linguistics.
- Weiting Tan, Kevin Heffernan, Holger Schwenk, and Philipp Koehn. 2023. [Multilingual representation distillation with contrastive learning](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1477–1490, Dubrovnik, Croatia. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. [Bicleaner AI: Bicleaner goes neural](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831, Marseille, France. European Language Resources Association.
- Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. [Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora](#). In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67, Vancouver, Canada. Association for Computational Linguistics.

A Illustration of the UnsupPSE approach

Figure 2 displays an adapted version of the segment-level processing of [Hangya and Fraser \(2019\)](#).

B SimAlign: Alignment strategy comparison

SimAlign provides three alignment strategies: Argmax, Itermax, and Match. While the Argmax strategy aligns tokens that are mutually most similar, Itermax applies this approach iteratively to increase the total number of alignments. Match, in contrast, considers the alignment task as a graph theory problem and computes a maximum-weight maximal matching. Hence, Argmax favors alignment precision, whereas Itermax and Match lead to higher recall.

C Example of token aggregation

Figure 3 shows an example of the subword aggregation to move from the subword to the word tokenization level. In this specific example, we use the SentencePiece tokenizer from Glot500-m ([Imani et al., 2023](#)) to tokenize the sentence.

D Stop-word list compilation

Obtaining predefined stop-word lists for low-resource languages is often challenging. A common approach to generate such lists involves processing a corpus of arbitrary sentences and calculating the term frequency of each word ([Manning et al., 2008](#)). Based on the premise that stop-words exhibit high frequency, the heuristic applied here considers the 300 most frequent words as stop-words. To further minimize the risk of including non-stop-words, each candidate word was manually verified against the annotated corpus provided by Universal Dependencies ([Nivre et al., 2020](#)).⁶ Words absent from the corpus were verified by translating them into a high-resource language using respective translation tools.⁷ For Chuvash, no annotated corpus was available. Therefore, we used a Russian stop-word list provided by NLTK ([Bird et al., 2009](#)) and translated the stop-words based on

⁶HSB corpus: https://github.com/UniversalDependencies/UD_Upper_Sorbian-UFAL, OCI corpus: https://github.com/UniversalDependencies/UD_Occitan-TTB

⁷Sotra for HSB: <https://sotra.app/>, Google Translate for OCI: <https://translate.google.com/?sl=oc&tl=en&op=translate>

a Chuvash–Russian dictionary. The pre-compiled stop-word lists for Chuvash, Upper Sorbian, and Occitan are made publicly available in the accompanying code repository (linked in Section 1).

E Post-processing hyperparameters

Hyperparameters	OCI-ES	CHV-RU	HSB-DE
PaSeMiLL static thresh.*	-1.00	-1.00	-1.00
window-size	7	7	7
min-segment-length	0.10	0.10	0.35
segment-threshold	0.03	0.06	0.20
filtering-threshold	0.45	0.34	0.38

Table 8: Best-performing hyperparameters of the post-processing pipeline for each language pair. *Static filtering threshold for the PaSeMiLL mining pipeline.

Table 8 summarizes the hyperparameters used in the filtering pipeline to obtain the results reported in Tables 2, 3, 4, and 5.

F fast_align: Performance comparison

Method / Configuration	Metrics		
	P	R	F
Occitan–Spanish (OCI-ES)			
Dice Threshold = 0.1	96.44	48.39	64.44
Dice Threshold = 0.01	94.06	70.69	80.72
Dice Threshold = 0.001	93.75	70.01	80.16
Chuvash–Russian (CHV-RU)			
Dice Threshold = 0.1	80.32	13.35	22.90
Dice Threshold = 0.01	67.85	20.43	31.40
Dice Threshold = 0.001	65.24	21.43	32.26
Upper Sorbian–German (HSB-DE)			
Dice Threshold = 0.1	92.86	04.33	8.28
Dice Threshold = 0.01	89.73	30.00	44.97
Dice Threshold = 0.001	85.54	30.37	44.82

Table 9: Comparison of fast_align results for BE-LOPSEM language pairs.

As previously noted, the alignments produced by fast_align ([Dyer et al., 2013](#)) tend to contain significant noise when applied to non- and partially parallel corpora. This noise primarily stems from the log-linear reparameterization underlying the fast_align model. Consequently, fast_align inherently prioritizes alignments that lie close to the diagonal between the source and target sentences ([Dyer et al., 2013](#)). In the context of non-parallel or partially parallel candidate pairs, this geometric

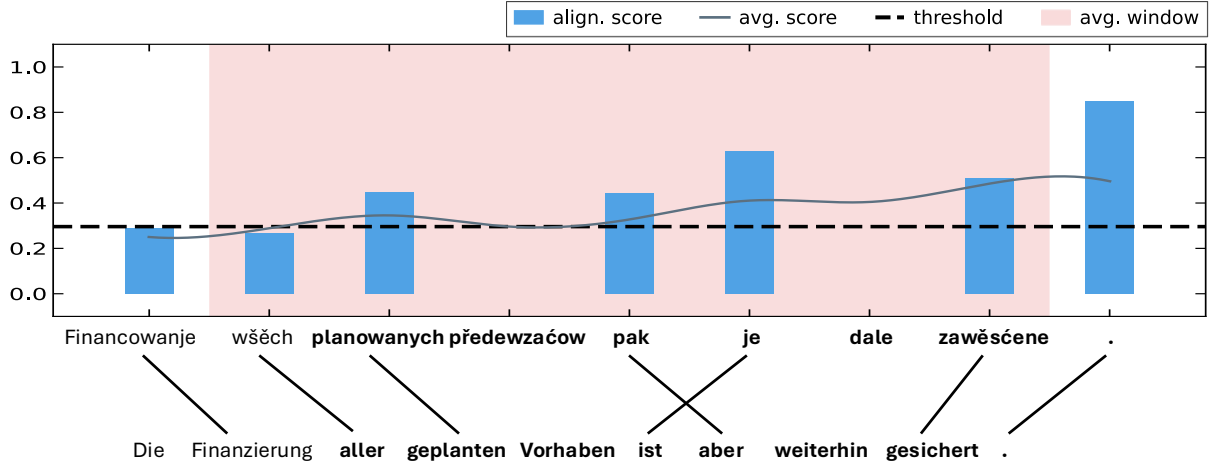


Figure 2: Visualization of the token alignment, averaging, and the subsequent building of contiguous segments for the HSB-DE language pair (adapted from [Hangya and Fraser, 2019](#)). The blue bars represent the cosine similarity values of aligned words. Contiguous segments are marked in **bold**. We use a window size of 7 for the averaging and a threshold of 0.3 for the contiguous segments. We present an actual translation pair for HSB-DE from BELOPSEM (‘However, the funding for all planned projects remains secure.’ in English).

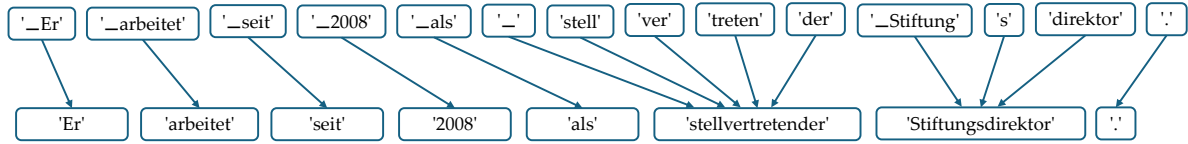


Figure 3: Example token aggregation according to the algorithm of the post-processing script.

bias leads to many erroneously aligned words. To mitigate this issue, a pre-filtering step is applied to the fast_align output before it is accessed by the post-processing pipeline. We apply the Dice Coefficient ([Dice, 1945](#)) to the obtained alignments to evaluate association strength solely based on alignment frequencies. Alignment pairs whose Dice Coefficient falls below a predefined threshold are classified as noise and subsequently discarded. The Dice Coefficient is defined as follows:

$$\text{Dice Coefficient} = \frac{2 \cdot h}{a + b} \quad (2)$$

In this context, h denotes the frequency with which the words w_1 and w_2 are aligned with each other across the entire fast_align output. Furthermore, a and b represent the total number of times w_1 and w_2 appear within the initial fast_align alignments, respectively.

Table 9 compares the final post-processing performance when the initial fast_align output is pre-filtered using different Dice Coefficient thresholds. The results demonstrate that a threshold of 0.1 is too strict, incorrectly filtering out a large portion of valid word alignments. During the subsequent

post-processing phase, this severe alignment reduction drastically shortens the detected parallel segments, ultimately resulting in a high number of false negatives.

G Ablation study: Punctuation handling

Punctuation handling entails the removal of the final token if it consists solely of punctuation marks. This approach is designed to mitigate the noise introduced by the terminal punctuation mark, which is frequently aligned with its corresponding counterpart. Through subsequent window-based averaging, this terminal mark can artificially inflate the overall similarity, particularly for short sentences. Although this configuration does not contribute to the overall performance of the post-processing pipeline, it nevertheless provides insights into the false positives present within the remaining candidate pairs. By comparing the results in Table 10, we can confirm that a significant amount of noise is introduced by the artifactual inflation of similarity scores caused by final punctuation tokens. While this approach filters a substantial amount of noise, it simultaneously results in a decline in recall.

Lang Pair	mean vect.	SW src	SW trg	Sim-Align	Punct. Handl.	P	R	F	Pair Num.
OCI-ES	d	No	No	A	No	84.97	83.05	84.00	1,424
	d	No	No	A	Yes	89.62	81.74	85.50	1,329
CHV-RU	d	Yes	Yes	I	No	55.83	44.73	49.67	1,200
	d	Yes	Yes	I	Yes	71.77	37.85	49.56	790
HSB-DE	s	No	Yes	A	No	80.26	49.73	61.41	1,859
	d	No	Yes	A	No	80.25	49.70	61.38	1,858
	d	No	Yes	A	Yes	91.71	43.90	59.38	1,436

Table 10: Ablation study results of the punctuation handling configuration.

H Statistical significance of the MT results

Table 11 reports the empirical p -values obtained from statistical significance testing on the machine translation results presented in Table 7. We conducted paired bootstrap resampling with 1,000 resamples using the sacreBLEU toolkit (Post, 2018). Across all four translation directions, both FT-sentences and FT-segments achieve statistically significant gains over the baseline (FT-mined) for both chrF2 and BLEU ($p < 0.001$). Furthermore, the direct comparison between FT-sentences and FT-segments indicates statistically significant differences ($p < 0.001$) in all settings, with the sole exception of the BLEU scores for both translation directions between Chuvash and Russian (CHV-RU).

Direction	Comparison	chrF2	BLEU
HSB → DE	FT-mined vs. FT-senten.	< 0.001	< 0.001
	FT-mined vs. FT-segm.	< 0.001	< 0.001
	FT-senten. vs. FT-segm.	< 0.001	< 0.001
DE → HSB	FT-mined vs. FT-senten.	< 0.001	< 0.001
	FT-mined vs. FT-segm.	< 0.001	< 0.001
	FT-senten. vs. FT-segm.	< 0.001	< 0.001
CHV → RU	FT-mined vs. FT-senten.	< 0.001	< 0.001
	FT-mined vs. FT-segm.	< 0.001	< 0.001
	FT-senten. vs. FT-segm.	< 0.001	0.0689
RU → CHV	FT-mined vs. FT-senten.	< 0.001	< 0.001
	FT-mined vs. FT-segm.	< 0.001	< 0.001
	FT-senten. vs. FT-segm.	< 0.001	0.0709

Table 11: Empirical p -values computed via paired bootstrap resampling (1,000 resamples) across all evaluated model pairs.

I Size-matched MT results

A potential confounding factor in the downstream MT results (Section 6.2) is the varying size of the corpora used to fine-tune the NLLB model (NLLB

Team et al., 2024). To isolate the impact of data quality from data quantity, we conducted an additional evaluation using size-matched corpora. We sub-sampled the FT-sentences and FT-segments corpora to match the target-side word count for FT-mined. For FT-sentences, we retained the top- k candidate pairs with the highest alignment similarity scores, whereas for FT-segments, we prioritized the longest available segment pairs.

Table 12 presents the downstream MT performance using these size-matched corpora. With the exception of the BLEU score difference between FT-segments with FT-sentences and FT-mined for RU→CHV, all observed differences between systems are statistically significant ($p < 0.01$). Crucially, the FT-sentences configuration continues to significantly outperform the FT-mined baseline. This confirms that our filtering pipeline yields intrinsically higher-quality parallel data, demonstrating that the improvements in Section 6.2 are not merely an artifact of larger data volume.

Conversely, the size-matched FT-segments configuration exhibits significant performance degradation compared to the baseline in several directions. This contrasts with our original unconstrained findings and suggests that while segment-level extraction is highly effective at scale, it potentially introduces context fragmentation when the overall training volume is severely restricted.

Direction	Setting	# Pairs	# Words	chrF2	BLEU
HSB → DE	zero-shot	-	-	35.60	9.20
	FT-mined	2,169	32,647	42.60	15.30
	FT-sentences	2,556	32,653	43.90*	16.40*
	FT-segments	1,353	32,659	41.20*	12.90*
DE → HSB	FT-mined	2,169	32,647	27.20	4.20
	FT-sentences	2,556	32,653	28.80*	5.10*
	FT-segments	1,353	32,659	25.80*	3.50*
CHV → RU	zero-shot	-	-	21.00	1.60
	FT-mined	7,789	55,866	21.00	2.50
	FT-sentences	10,068	55,898	25.80*	3.70*
	FT-segments	958	55,912	22.50*	2.00*
RU → CHV	FT-mined	7,789	55,866	20.00	1.40
	FT-sentences	10,068	55,898	20.70*	1.70*
	FT-segments	958	55,912	18.20*	1.60

Table 12: Downstream machine translation performance of the NLLB model for HSB-DE and CHV-RU. ‘# Pairs’ indicates the number of sentence or segment pairs used for fine-tuning. ‘# Words’ represents the number of words in the high-resource language. Superscript * denotes statistically significant difference compared to ‘FT-mined’, while † indicates a significant difference between ‘FT-sentences’ and ‘FT-segments’ (p -value < 0.01).