AlignFix: A Tool for Parallel Corpora Augmentation and Refinement

Samuel Frontull and Simon Haller-Seeber
Department of Computer Science, University of Innsbruck, Austria
{samuel.frontull,simon.haller-seeber}@uibk.ac.at

Abstract

High-quality datasets are crucial for training effective state of the art machine translation systems. However, due to the data-intensive nature of these systems, they have to be trained on large amounts of text that can easily go beyond the scope of full human inspection. This makes the presence of noise that can degrade overall system performance a frequent and significant issue. While various approaches have been developed to identify and select only the highest-quality training examples, this is undesirable in scenarios where resources are limited. For this reason, we introduce AlignFix, an open-source tool for augmenting data, identifying and correcting errors in parallel corpora. Leveraging word alignments, AlignFix extracts consistent phrase pairs, enabling targeted replacements that can improve the dataset quality. Besides targeted replacements, the tool enables contextual augmentation by duplicating sentences and allowing users to substitute words with alternatives of their choice. The tool maintains and updates the underlying word alignments, thereby avoiding the costly recomputation. AlignFix runs locally in the browser, requires no installation, and ensures that all data remains entirely on the client side. It is released under Apache 2.0 license, encouraging broad adoption, reuse, and further development. A live demo is available at https://ifi-alignfix.uibk.ac.at.

1 Introduction

High-quality, carefully curated datasets are critical for the development of reliable machine translation (MT) systems. In an ideal scenario, only fully manually verified data would be available. However, neural MT systems are highly data-intensive (Koehn and Knowles, 2017; Gordon et al., 2021), necessitating the collection of as many texts as possible for training. Although modern architectures have enabled transfer learning for scenarios with limited resources (Zoph et al., 2016), a sufficient amount of training data must still be accumulated (Gu et al., 2018).

For machine translation, the so-called *contextual augmentation* (Kobayashi, 2018; Wu et al., 2019; Gao et al., 2019) is an established technique for data augmentation and extends existing corpora by reusing existing sentences and replacing words in them. This technique is particularly effective for enhancing lexical coverage and ensuring the representation of rare words. However, this method requires a solid data foundation and relies on language models that provide sensible replacements, which are usually not available in low-resource scenarios.

A significantly more accessible and effective alternative is *back-translation* (Sennrich et al., 2016) which involves translating available monolingual target-language text into the source language using an auxiliary model. This approach can provide the (synthetic) parallel training data necessary to leverage state-of-the-art neural architectures for MT. In practice, however, datasets often become unwieldy when scaled to meet these requirements, resulting in a diminished insight into the data.

Synthesised data often contains numerous errors that can negatively impact the overall quality of a machine translation system (Hoang et al., 2018). Noisy data can for example lead to erroneous translations or hallucinations (Khayrallah and Koehn, 2018; Guerreiro et al., 2023). Consequently, it is crucial to filter and clean such datasets. Several tools have been developed for this purpose (Bogoychev et al., 2023; Zaragoza-Bernabeu et al., 2022; Aulamo et al., 2020). However, these tools primarily focus on data filtering, retaining only the highest-quality translation pairs and discarding the remainder. In contexts where data is scarce, aggressive filtering is not always desirable (Marashian et al., 2025). In such cases, it is preferable to identify and correct errors.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 215–224

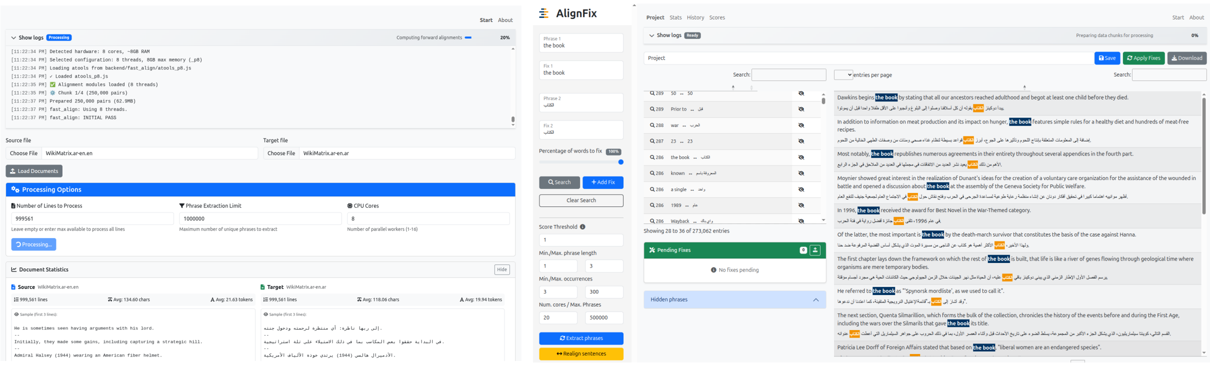

Figure 1: Left: Screenshot showing text alignment and phrase extraction. Right: Project view displaying extracted phrases. Both images are taken while working with the WikiMatrix Arabic–English corpus.

ALIGNFIX addresses this need by providing a tool for refining and augmenting parallel datasets through the extraction of aligned words and phrases, allowing for targeted, contextual interventions. The process works as follows:

(i) the texts are tokenized to separate words from punctuation,

(ii) (symmetric) word alignments are computed,

(iii) phrase pairs that are translations of one another are extracted.

Fixes can be specified for these phrase pairs and applied selectively, allowing targeted adjustments only where intended. The tool maintains and incrementally updates the underlying word alignments when fixes are applied, thereby avoiding the costly recomputation of the alignments. The tool is designed to handle datasets up to one million samples efficiently and offers a user-friendly web interface. Figure 1 presents two screenshots of the interface. The left image illustrates text alignment and phrase extraction, while the right image shows the project view with the extracted phrases. Both screenshots were captured during work with the WikiMatrix Arabic–English corpus (Schwenk et al., 2021).

The primary use case for ALIGNFIX is thus to refine parallel corpora. However, it can also be used for other scenarios that require word-level alignment, such as augmenting corpora with glossary-enforced training data. In this work:

- we adapt and compile existing word-alignment and phrase-extraction methods so that they run directly in the browser, making them executable for everyone without installation, compilation, or technical expertise;
- we present ALIGNFIX, an open-source tool that allows for targeted data augmentation and refinement of parallel corpora and demonstrate its performance on different datasets;
- we demonstrate its practical utility in a low-resource domain scenario using two novel datasets of meteorological forecasts, which we make publicly available.

ALIGNFIX is available at `https://ifi-alignfix.uibk.ac.at` and is demonstrated in a supplementary video[1]. The source code[2] is provided under the Apache 2.0 open-source licence.

2 Related Work

Several toolkits have been developed to automate the cleaning and preparation of bitexts. `OpusCleaner` and `OpusTrainer` (Bogoychev et al., 2023) are widely adopted open-source toolkits that streamline downloading, preprocessing, and mixing data for large-scale neural MT. Similarly, `OpusFilter` (Aulamo et al., 2020) offers a modular toolbox for filtering, language identification, and alignment, allowing users to chain custom heuristic filters. For noise detection, `Bicleaner` and its successor `Bicleaner AI` (Zaragoza-Bernabeu et al., 2022) identify and discard noisy sentence pairs. While Bicleaner relies on heuristics, Bicleaner AI utilizes transformer-based models for a more accurate text classification. However, these approaches primarily function as filters. In low-resource scenarios, as highlighted by Marashian et al. (2025), data scarcity makes the rejection of "imperfect" sentence pairs undesirable. Discarding data that

[1] `https://youtu.be/F_7fyWc4vZo`
[2] `https://github.com/alignfix/alignfix`

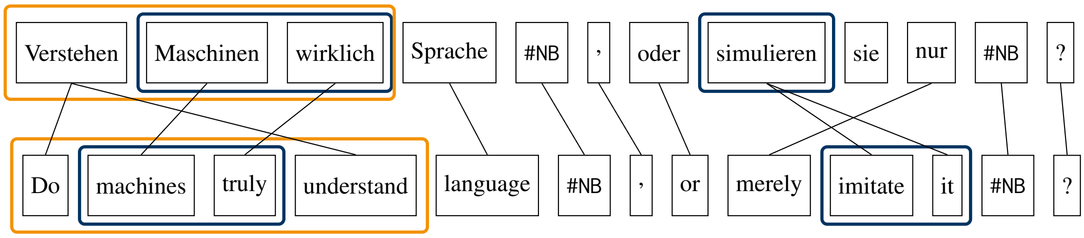

Figure 2: Alignment between a German and an English sentence with the consistent phrase pairs up to length 3.

contains recoverable errors can further starve an already data-poor system.

Beyond simple filtering, some tools provide functionalities to clean and fix problematic elements in corpora. `Bifixer`, part of the Bitextor project, focuses on technical repairs such as fixing encoding errors and removing near-duplicate sentence pairs (Ramírez-Sánchez et al., 2020). For the creation and management of alignments, `SentAlign` (Steingrímsson et al., 2023) utilizes LaBSE embeddings to identify semantically similar sentence pairs, employing dynamic programming for optimal alignment recovery. Once corpora are created, tools like `InterText` (Vondřička, 2014) provide a flexible editor for managing and manually aligning parallel texts.

While the aforementioned tools excel at either bulk filtering or sentence-level management, there is a lack of lightweight tools designed for corpus refinement and *contextual* word augmentation.[3] `AlignFix` addresses this by leveraging word alignments to allow targeted, manual replacements without discarding samples, thereby preserving valuable training data.

3 AlignFix

In this section, we describe our method and provide implementation details. In the first part, we describe the steps involved in extracting the phrase pairs from parallel corpora. In the second part, we discuss how fixes can be applied and how samples are augmented.

3.1 Method Overview

In this section, we explain the individual steps of tokenization, alignment, and phrase extraction. All three components are implemented in (parallelized) C and compiled to WebAssembly (WASM), enabling efficient execution directly in the browser. The resulting tokenized texts, word alignments, and extracted phrase pairs are persisted in an in-browser SQLite database.

Tokenization The computation of the word-alignments, requires tokenized texts (tokens are separated by blanks) as input. Therefore, as first step, we tokenize each sentence by explicitly separating punctuation from the surrounding text. Whenever punctuation is attached directly to a word without an intervening space, we insert a dedicated non-blank marker token (`#NB`) to ensure that the resulting tokenized text is reversible. For example, the sentence *The corpus is small, but valuable.* is tokenized to *The corpus is small* `#NB` *, but valuable* `#NB` .. This allows us to later reconstruct the original text (with possible fixes).

Word-Alignments To compute word alignments, we rely on the tokenized texts as produced in the preprocessing step. For the computation of the word-alignments, we use `fast_align` (Dyer et al., 2013). Figure 2 shows an example of word-alignments between two sentences.

We adapted the original C++ implementation and compiled it to WebAssembly using `emcc` (Zakai, 2011). This required several modifications: (i) we exposed the `main` function and key entry points to `emcc` so they could be invoked directly from JavaScript; (ii) we replaced the original OpenMP-based parallelisation with a WebAssembly-compatible setup using `pthreads`, enabling multi-threaded execution inside the browser; and (iii) we adjusted the build configuration to allow the model parameters to be loaded from in-memory buffers rather than from the local file system. These changes produced a browser-executable alignment tool with efficient parallel processing , allowing us to run alignment entirely in the browser. Symmetrization is carried out using `atools`, which we likewise compiled to WebAssembly following the same procedure.

[3] `OpusTrainer` also offers data augmentation, but focuses on surface-level text manipulations (e.g., casing, all-caps) to improve model robustness rather than contextual refinement.

Phrases Extraction Using the word alignments, we extract all sequences of aligned words up to a predefined length and record every occurrence of these phrases. To do this, we have implemented the method introduced in Och et al. (1999) and refined in Koehn et al. (2003) to extract *consistent* phrase pairs in C++ and compiled it to WebAssembly using `emcc`, enabling parallel execution via `pthreads`. This allows fast retrieval and inspection of their occurrences in the corpus. To make this process memory-efficient, the corpus is processed in batches, with batch sizes configurable based on available system memory.

We only collect *consistent* phrase pairs as they can be fully replaced without affecting other words that may occur in between otherwise. For the example illustrated in Figure 2, beside the single word pairs, the extracted phrases up to a maximum length of three would be:

1. ⟨Maschinen; machines⟩, ⟨wirklich; truly⟩, ⟨Sprache; language⟩, ⟨oder; or⟩, ⟨nur; merely⟩
2. ⟨Maschinen wirklich; machines truly⟩, ⟨simulieren; imitate it⟩.
3. ⟨Verstehen Maschinen wirklich; Do machines truly understand⟩

We do not include ⟨Maschinen wirklich Sprache; machines truly understand language⟩, because the word understand is not aligned to any word in Maschinen wirklich Sprache. We trim punctuation and non-blank symbols (e.g. we treat ⟨Sprache #NB; language #NB⟩ as ⟨Sprache; language⟩). We also remove pairs only consisting of punctuation symbols, e.g. ⟨?; ?⟩, ⟨,; ,⟩ or ⟨#NB ,; #NB ','⟩.

Managing Extraction Scale There is one drawback of extracting all phrases in the corpus: the number of extracted phrases grows rapidly. For example, in a corpus of 100k sentences, the number of phrase pairs can easily exceed one million with a maximum phrase length of three. Storing all of them would introduce substantial overhead. Therefore, we allow the user to specify an upper limit on the number of phrase pairs to collect (default: 500k). Based on the maximum phrase length, we determine an appropriate batch size for processing the corpus, so that we can guarantee to stay below a peak memory usage of 4GB[4]. After each batch, we check whether the number of collected phrase pairs exceeds the user-defined limit. If so, we prune phrase pairs with a single occurrence within the processed batch. If the limit is still exceeded, we iteratively remove pairs with two occurrences, three occurrences, and so forth, until the total number of phrase pairs falls below the threshold. We then proceed with the next batch.[5] Users who are interested in phrase pairs that rarely occur can still search for them directly in the full corpus.

[4]See discussion on memory considerations in Section 4

3.2 Corpus Augmentation and Refinement

This section describes the implemented features for corpus augmentation and refinement.

Data Augmentation ALIGNFIX supports contextual data augmentation. The user can duplicate existing sentence pairs and selectively replace aligned words to generate new training examples. For instance, whenever the word *car* occurs, it can be substituted with *automobile*, along with the corresponding replacement in the target language. Similarly, even substitutions that change the meaning but still yield coherent sentences can be applied when appropriate. For example, in a sentence such as *"She touched her ear."* the word *ear* may be replaced with *nose* to form *"She touched her nose."* While not all contexts support such substitutions, ALIGNFIX enables users to perform them in a controlled manner, thereby enriching the corpus with additional valid sentence pairs. This enables controlled lexical diversification and allows users to introduce examples for terms that are underrepresented or entirely absent from the original corpus.

Refinement Word alignments are leveraged to enable users to correct phrases in both the source and target sentences. In ALIGNFIX, these corrections can be applied either to every occurrence of a given phrase, or to a selected subset of occurrences across the corpus. For instance, based on the phrase pairs extracted in Figure 2, a user could choose to replace *Do machines truly understand* with *Can machines understand* throughout the corpus, wherever it aligns with *Verstehen Maschinen wirklich*. Computing word alignments is computationally expensive. Therefore, ALIGNFIX preserves alignment consistency when replacements are applied. In cases where a single token is replaced by multiple tokens, the original aligned to-

[5]In the worst case, this procedure may discard (rare but important) phrase pairs that would have appeared `corpus_size / batch_size` times across the entire dataset but this pruning is necessary to avoid hitting memory limits.

Corpus	Size	Cores	Tokenization (T)		Alignment (A)		Phrase Extraction (P)		DB Insert	Efficiency (s / 1k lines)		
			Time (s)	Mem (MB)	Time (s)	Mem (MB)	Time (s)	Mem (MB)	Time (s)	T	A	P
S_{6k}	6k	8	0.67	9.70	13.16	1.10	7.57	125.00	1.22	0.11	2.19	1.26
S_{6k}	6k	16	0.66	11.44	7.16	7.51	5.19	19.94	0.57	0.11	1.15	0.84
D_{100k}	100k	8	5.39	121.96	223.92	108.61	66.90	49.57	13.34	0.05	2.24	0.67
D_{100k}	100k	16	3.14	113.07	103.94	109.56	86.24	499.80	24.48	0.03	1.04	0.86
W_{418k}	418k	8	26.07	298.93	579.44	310.97	412.40	984.75	44.02	0.06	1.39	0.99
W_{418k}	418k	16	9.92	358.96	326.05	9.54	306.72	872.06	68.24	0.02	0.78	0.73
W_{1M}	1M	8	78.33	843.10	2073.67	499.00	1329.66	2082.20	162.34	0.08	2.07	1.33
W_{1M}	1M	16	40.99	1005.00	1182.35	384.35	985.88	2332.95	133.19	0.04	1.18	0.99

Table 1: Benchmark results for tokenization, alignment, and phrase extraction across corpora, including efficiency normalized per 1k sentence pairs.

kens are distributed across the new tokens to maintain the alignment structure.[6]

Even after the pruning of phrases described above, the remaining set of phrase pairs may still be large. Therefore, to facilitate the identification of potential error candidates, ALIGNFIX provides two additional filtering mechanisms that can substantially reduce the number of phrase pairs: (i) ignore known phrase pairs. (ii) filter based on translation quality scores. In (i), the user may upload a list of phrase pairs to be excluded from extraction which could for example be a list of verified translations. In (ii), the user may upload quality scores for the sentence pairs in the corpus. Scores should range from 0 (low-quality translation) to 1 (high-quality translation) – for instance, 0 for back-translated data and 1 for expert translations. The user can then define a threshold: only phrase pairs occurring exclusively in translations with a score below the threshold are retained.[7]

4 Experimental Setup

The aim of our experiments is to systematically evaluate the effectiveness of the tool introduced in this work, both with respect to its operational performance and the impact it can have on downstream machine translation.

4.1 Tool Performance

To assess the performance, we conducted experiments across multiple corpora and computing environments. In Table 1 we report the results.

Benchmark The benchmark suite covers three publicly available corpora of different sizes and linguistic characteristics. S_{6k} represents the 6k-sentence Seed corpus (Maillard et al., 2023) for Italian (Ferrante, 2024) and serves as a controlled small-scale reference for evaluating baseline throughput on a low-volume dataset. D_{100k} corresponds to a 100k-sentence heterogeneous general-domain corpus for Uzbek–Karakalpak (Mamasaidov and Shopulatov, 2024), enabling analysis on medium-sized data. To assess performance on substantially larger material, W_{418k} uses a 418k-sentence subset of the *WikiMatrix* German–Spanish corpus (Schwenk et al., 2021). Finally, W_{1M} contains one million sentence pairs from the *WikiMatrix* Arabic–English corpus (Schwenk et al., 2021), selected as a large-scale and cross-family dataset to stress-test the tool under substantial data volume. Together, these corpora enable a comprehensive evaluation of runtime, memory usage, and throughput across variation in size and languages.

Hardware and Performance The performance was evaluated on two systems running Chromium v142. The primary Mini-PC features a 12th Gen Intel® Core™ i9-12900H (20 cores) with 64 GB DDR4-3000 memory, used for 16-thread tests with 16 GB WASM memory and a 4 GB JS heap. A lower-resource laptop[8] with an 8th Gen Intel® Core™ i7-8565U (4 cores, 8 threads) and 40 GB DDR4-2667 memory was used for 8-thread tests with 8 GB WASM memory and a 4 GB JS heap. Across both machines, the runtime data demonstrate that the full pipeline scales efficiently with available parallelism, and that large corpora are processable within browser constraints.

[6]This heuristic alignment repair strategy provides a practical approximation; more accurate alignments could be obtained by recomputing them after each replacement.

[7]ALIGNFIX provides experimental metrics to estimate translation quality. These metrics are currently under development and have not yet been fully validated; their evaluation and refinement are planned as part of future work.

[8]Due to OS-level power scaling, the effective CPU frequency (and thus execution time) may vary on battery or under background load.

Scalability and Efficiency The efficiency values in Table 1 show that processing time (in seconds) per 1k sentence pairs decreases as corpus size increases, demonstrating favorable scaling of the pipeline. From the 6k corpus to the 100k corpus, tokenization time drops from `0.11s` to `0.05s` per 1k lines, while alignment throughput remains effectively constant, changing only slightly from `2.19s` to `2.24s` despite the larger dataset. Phrase extraction also becomes more efficient at scale, decreasing from over `1s` per 1k lines on small data to `0.67s` for the 100k corpus and reaching `0.73s` per 1k lines on the 418k corpus, before stabilizing on the 1M corpus. These results indicate that one-time initialization and model-loading overheads are quickly amortized, and that the pipeline benefits substantially from multithreaded execution.

Memory Considerations The memory measurements in Table 1 report only JS heap usage (`usedJSHeapSize`), which is managed by V8's garbage collector. WebAssembly linear memory, allocated separately, is not included; although its size can be obtained via `WebAssembly.Memory.buffer.byteLength`, including it would mix separate memory regions and could be misleading. The WASM modules were compiled with a maximum linear memory of 1 GB per CPU core, matching the 8-thread (8 GB) and 16-thread (16 GB) configs used in our experiments.

Across all experiments, JS heap usage remained below 2.5 GB, safely within the browser's 4 GB limit, with minor fluctuations due to garbage collection. Combined WASM+JS memory usage therefore stayed below the effective upper bounds of 12 GB (4+8) or 20 GB (4+16), even on the largest 1M-sentence corpus. Overall, memory consumption grows moderately with corpus size.

4.2 Impact of Targeted Corrections

To illustrate the applicability of ALIGNFIX in a realistic low-resource scenario, we conducted experiments on weather forecast texts provided by the *Amt für Meteorologie und Lawinenwarnung* of the Autonomous Province of Bolzano – South Tyrol[9]. The corpus consists of 689 parallel Ladin (Val Badia)–German (`VB–DE`) reference translations[10] and additional 15,969 `VB`-only weather forecast texts[11]. This setup reflects a typical low-resource condition, where the lack of parallel corpora necessitates the synthesis of training data through backtranslation.

Model	BLEU	COMET
Ladin (Val Badia) → German		
`gemini-2.5-flash-lite`	15.9±1.2	67.0
`Helsinki-NLP/opus-mt-it-de`		
fine-tuned with backtranslations	17.0±1.2	67.8
+ 138 fixes with ALIGNFIX	**18.6±1.2**	69.6
German → Ladin (Val Badia)		
`Helsinki-NLP/opus-mt-de-it`		
fine-tuned with backtranslations	30.5±1.6	55.7
+ 138 fixes with ALIGNFIX	**32.3±1.5**	56.6

Table 2: Comparison of translation quality ($\mu\pm$ 95% CI) for German–Ladin weather forecasts, highlighting the gains achieved by applying 138 targeted corrections.

Data Augmentation via Backtranslation We generated a synthetic parallel dataset by translating the 15,969 Ladin monolingual texts into German using `Gemini 2.5 Flash-Lite` (Comanici et al., 2025) in a zero-shot setting.[12] Following the backtranslation paradigm, we then fine-tuned a `DE` → `VB` model on this synthetic corpus. As no pre-trained German–Ladin model is available and Ladin is closely related to Italian, we used the `Helsinki-NLP/opus-mt-de-it` model (Tiedemann et al., 2024; Tiedemann and Thottingal, 2020) as base model for this experiment. The model was trained for up to 20 epochs with a batch size of 8 and learning rate $2 \cdot 10^{-5}$, with early stopping set to 3 epochs.

Targeted Corrections After establishing this baseline, we used ALIGNFIX to identify and correct systematic errors in the German backtranslations produced by the large language model (LLM). In total, we applied 138 targeted phrase-level fixes and fine-tuned the model on this refined data (with the same configuration).

To quantify the scale of the applied corrections: the 138 fixes modified 6,677 of the 15,969 synthetic sentences (41.8%). In total, ALIGNFIX introduced 56,906 character-level edits, corresponding to an average of 85 edits per changed sentence and an overall edit intensity of 37.3% relative to the

[9]Datasets released by the authors with permission of the *Amt für Meteorologie und Lawinenwarnung*.

[10]`https://huggingface.co/datasets/sfrontull/south-tyrol-weather-lld-deu`

[11]`https://huggingface.co/datasets/sfrontull/south-tyrol-weather-lld`

[12]Prompt: Translate the following sentence from Ladin to German: <Ladin_Text>

original text. These figures highlight that a moderate number of targeted interventions (requiring roughly 1–2 hours of manual effort) can propagate broadly across a domain corpus, producing substantial systematic improvements. A complete list of the applied fixes is provided in Appendix A.

Results Table 2 reports the BLEU and COMET scores for the evaluated models. The BLEU scores were computed using `sacreBLEU` (Post, 2018). We employed paired bootstrap resampling (`--paired-bs`) to assess statistical significance; values in bold denote a significant improvement over the baseline. The COMET scores were computed using the `Unbabel/wmt22-comet-da` (Rei et al., 2022) model. The results clearly demonstrate the positive effect of our interventions on translation quality. Fine-tuning on the backtranslated data results in 30.5 BLEU. Applying targeted corrections with ALIGNFIX increases performance to 32.3 BLEU (+1.8 BLEU). Despite Ladin being unsupported by COMET, the 0.9 increase suggests improvements in semantic adequacy as well.

We also examined the effect of these fixes in the opposite translation direction (Ladin → German). Fine-tuning on the synthetic corpus already improves over the zero-shot Gemini baseline (+1.1 BLEU and +0.8 COMET). The refined corpus (in this case, with target-side corrections) yields further improvements, reaching 18.6 BLEU and 69.6 COMET (additional +1.6 BLEU and +1.8 COMET).

5 Conclusion

We presented ALIGNFIX, a tool for improving parallel corpora by leveraging word alignments to propagate corrections consistently across sentence pairs. ALIGNFIX enables users to modify individual tokens or phrases while automatically maintaining alignment integrity, even when a single token is replaced by multiple tokens. Through its combination of browser-executable algorithms and phrase-based repair operations, the system offers a flexible, scalable, and user-friendly framework for enhancing translation corpora across a wide range of practical scenarios.

Our experiments highlight an important mechanism in low-resource machine translation where training data is synthesized. Errors in the synthetic texts can systematically remove or distort domain-specific terminology. If key terms are mistranslated or omitted during backtranslation, they never appear aligned with their correct counterparts in the synthetic parallel data. As a result, the model fails to learn these correspondences and may later hallucinate or substitute more frequent but incorrect alternatives at inference time. By restoring correct terminology and phrase structure on the synthetic source side, ALIGNFIX allows to reintroduce these missing lexical links, strengthening the learned cross-lingual mapping and reducing errors in translation.

Future Work We consider the automated identification of potential errors to be a crucial feature. While the current functionality supports user-defined lists of phrase-pairs to exclude (e.g., to filter out correct pairs that do not require review), this is not a scalable solution. Potential errors could also be detected intrinsically. In future work, we would like to explore such methods and provide users with suggestions for possible fixes to substantially reduce the amount of manual work required.

Our current experiments and implementation support corpora of up to approximately one million sentence pairs. Larger datasets may exceed the memory limitations of the underlying database and browser execution environment. In future work, we further aim to improve memory prediction, adapt batch sizing to corpus characteristics, and optimize I/O and storage efficiency (e.g., OBFS) to handle even larger corpora.

Limitations

While ALIGNFIX is largely language-agnostic, the current implementation relies on whitespace-based tokenization and existing word alignment tools, which limit direct applicability to languages without explicit word boundaries (e.g., Chinese, Japanese, Thai). Lightweight, WASM-compatible tokenization strategies could be integrated to support scripts without whitespace segmentation, applying language-specific tokenizers only when necessary while preserving a unified, aligner-compatible output format.

ALIGNFIX assumes pre-aligned parallel data. This design choice reflects the primary target use case, where alignment is implicitly provided by construction. In scenarios where sentence alignment is unavailable or noisy, additional preprocessing is required. Moreover, the effectiveness of this tool depends critically on the quality of word alignments. If a corpus is too small to support robust statistical alignment, the approach may fail to produce satisfactory results.

Acknowledgments

This work was carried out as part of the research project *Intelligent Writing Assistant for Ladin* at the University of Innsbruck, in collaboration with the Ladin Cultural Institute “Micurá de Rü”. We thank the reviewers for their valuable comments and the Amt für Meteorologie und Lawinenwarnung of the Autonomous Province of Bolzano/Bozen – South Tyrol for providing the data.

References

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A Configurable Parallel Corpus Filtering Toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 20–26. Association for Computational Linguistics.

Nikolay Bogoychev, Jelmer van der Linde, Graeme Nail, Barry Haddow, Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Lukas Weymann, Tudor Nicolae Mateiu, Jindřich Helcl, and Mikko Aulamo. 2023. OpusCleaner and OpusTrainer: Open Source Toolkits for Training Machine Translation and Large Language Models. *arXiv preprint arXiv:2311.14838*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

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.

Edoardo Ferrante. 2024. A High-quality Seed Dataset for Italian Machine Translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 567–569, Miami, Florida, USA. Association for Computational Linguistics.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft Contextual Data Augmentation for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. 2021. Data and Parameter Scaling Laws for Neural Machine Translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal Neural Machine Translation for Extremely Low Resource Languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative Back-Translation for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the Impact of Various Types of Noise on Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Jean Maillard, Cynthia Gao, Elahe Kalbassi, Kaushik Ram Sadagopan, Vedanuj Goswami, Philipp Koehn, Angela Fan, and Francisco Guzmán.

2023. Small Data, Big Impact: Leveraging Minimal Data for Effective Machine Translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2740–2756, Toronto, Canada. Association for Computational Linguistics.

Mukhammadsaid Mamasaidov and Abror Shopulatov. 2024. Open Language Data Initiative: Advancing Low-Resource Machine Translation for Karakalpak. In *Proceedings of the Ninth Conference on Machine Translation*, pages 606–613, Miami, Florida, USA. Association for Computational Linguistics.

Ali Marashian, Enora Rice, Luke Gessler, Alexis Palmer, and Katharina von der Wense. 2025. From Priest to Doctor: Domain Adaptation for Low-Resource Neural Machine Translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7087–7098, Abu Dhabi, UAE. Association for Computational Linguistics.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

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, Belgium, Brussels. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and Bicleaner: Two Open-Source Tools to Clean Your Parallel Data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT 2020)*, pages 1875–1879. European Association for Machine Translation.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2023. SentAlign: Accurate and Scalable Sentence Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 256–263. Association for Computational Linguistics.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raúl Vázquez, and Sami Virpioja. 2024. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, 58(2):713–755.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Pavel Vondřička. 2014. Aligning Parallel Texts with InterText. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1875–1879. European Language Resources Association (ELRA).

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT Contextual Augmentation. In *Computational Science – ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV*, page 84–95, Berlin, Heidelberg. Springer-Verlag.

Alon Zakai. 2011. Emscripten: an LLVM-to-JavaScript compiler. In *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, OOPSLA '11, page 301–312, New York, NY, USA. Association for Computing Machinery.

Jaume Zaragoza-Bernabeu, Marta Bañón, Gema Ramírez-Sánchez, and Sergio Ortiz-Rojas. 2022. Bicleaner AI: Bicleaner Goes Neural. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 824–831.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Detailed Phrase-Level Refinements

Table 3 presents all 138 targeted interventions, their frequency in the backtranslated data, and a comparison of baseline German output with our refined translations. Note, for example, the different translations of the Ladin word *niores* (clouds) in the German texts hallucinated by the LLM, ranging from *Blumen* to *Mädchen*.

	Ladin (VB)	Original (DE)	Fixed (DE)	#		Ladin (VB)	Original (DE)	Fixed (DE)	#
1	indô	wieder	erneut	915	70	Tres plü variabl	Drei variabel	Zunehmend unbeständig	60
2	sovënz	oft	meist	694	71	cuntra le Südtirol	gegen Südtirol	Richtung Südtirol	60
3	danmisdé	am Nachmittag	am Vormittag	431	72	niores	Mädchen	Wolken	59
4	presciun bassa	Tiefdruckgebiet	Tief	370	73	Tres plü da nio	Drei mehr als nichts	Zunehmend bewölkt	59
5	niores	Nebel	Wolken	367	74	niores	Nächte	Wolken	58
6	temperatöres mascimes	Höchsttemperaturen	Höchstwerte	346	75	Sorëdl y niores	Sonne und Blumen	Sonne und Wolken	58
7	Da doman	Von morgen an	In der Früh	316	76	indlunch	wieder	überall	58
8	Sorëdl y niores	Sonnenschein und Blumen	Sonne und Wolken	302	77	Da sorëdl	Von der Sonne	Sonnig	57
9	en pert	teilweise	teils	279	78	manco tömies	weniger dicht	weniger feucht	55
10	Domisdé	Heute	Nachmittag	277	79	Da sorëdl y cialt	Von Sonne und Wärme	Sonnig und warm	54
11	bel	schön	freundlich	269	80	da doman	von morgen	in der Früh	54
12	instabil	instabil	unbeständig	259	81	sorëdl	oben	Sonne	53
13	en gran pert	größtenteils	überwiegend	241	82	süd dla provinzia	Süden der Provinz	Süden des Landes	53
14	Na presciun alta	Ein hoher Druck	Ein Hoch	238	83	döt sarëgn	ganz klar	wolkenlos	52
15	pert dla provinzia	Teil der Provinz	Teil des Landes	236	84	dadoman	morgen	in der Früh	51
16	limit dla nëi	Schneegrenze	Schneefallgrenze	233	85	vënt da nord	Wind aus Norden	Nordwind	50
17	La presciun alta	Der hohe Druck	Das Hoch	217	86	Dantadöt sorëdl	Gib mir die Sonne	Überwiegend sonnig	49
18	da nio	von Schnee	bewölkt	209	87	gnanca na niora	nicht einmal eine Wolke	wolkenlos	43
19	por intant	vorerst	vorübergehend	206	88	Valgamia	Wir gehen	Recht	42
20	Sön munt	Auf dem Berg	Auf den Bergen	203	89	Plülere sorëdl	Die Schwestern	Recht sonnig	42
21	Da sorëdl y da nio	Von Sonne und von Schnee	Sonne und Wolken	202	90	niores zënza faziun	Wolken ohne Niederschlag	harmlose Wolken	38
22	mascimes	Höchsttemperaturen	Höchstwerte	199	91	Sön la	Auf der Alpenhauptkamm	Am Alpenhauptkamm	37
23	niores	Blumen	Wolken	197	92	Sön la Ciadëna	Auf der Alpenhauptkamm	Am Alpenhauptkamm	37
24	Ciadëna	Alpenkette	Alpenhauptkamm	184	93	niores a gröm	Wolken	Quellwolken	37
25	i crëps	den Gipfeln	den Bergen	179	94	Dër da nio	Sehr gut	Sehr bewölkt	37
26	Dadoman	Morgen	In der Früh	178	95	Da nio	Von nichts	Bewölkt	36
27	Tres	Drei	Zunehmend	177	96	dantadöt	hauptsächlich	überwiegend	34
28	Da doman	Von morgen	In der Früh	174	97	Ciadëna zentrala dles	der zentralen Alpenhauptkamm	dem Alpenhauptkamm	33
29	la Ciadëna	Kette	Alpenhauptkamm	172	98	dantadöt	vor allem	überwiegend	32
30	da sorëdl	von der Sonne	sonnig	171	99	Plü variabl	Mehr variabel	Wechselhafter	30
31	Domisdé	Morgen	Am Nachmittag	166	100	meste	meistens	mild	30
32	manco da nio	weniger von nichts	weniger bewölkt	161	101	sön la	auf der Alpenhauptkamm	am Alpenhauptkamm	29
33	bonamënter	meist	voraussichtlich	144	102	sön la Ciadëna	auf der Alpenhauptkamm	am Alpenhauptkamm	29
34	Da sorëdl y	Von der Sonne und	Sonnig und	137	103	gnanca na niora	nicht einmal eine Stunde	wolkenlos	29
35	Domisdé	Vormittags	Nachmittags	132	104	zënza faziun	ohne Auflösung	harmlos	28
36	Sön la	Auf der zentralen Alpenhauptkamm	Am Alpenhauptkamm	132	105	vignitant	bald	zeitweise	28
37	moscedoz	Mix aus	Mischung aus	131	106	Danmisdé	Morgen	Am Vormittag	28
38	Domisdé	Heute Morgen	Am Nachmittag	130	107	condiziuns	Bedingungen	Verhältnisse	27
39	te tröc posć	an vielen Orten	verbreitet	129	108	naota	mehr	zunächst	26
40	tröp	zu viel	viel	129	109	niores a gröm	Haufen	Quellwolken	25
41	meste	Nebel	mild	128	110	dër meste	sehr traurig	sehr mild	24
42	arbassa	sinken	gehen zurück	125	111	Tröpes niores	Kleine Tropfen	Viele Wolken	24
43	Domisdé	Übermorgen	Am Nachmittag	122	112	Dër da sorëdl	Sehr von Sonne	Sehr sonnig	22
44	I valurs mascimai	Die maximalen Werte	Höchstwerte	117	113	Ciarü alt	Schau hoch	Hochnebel	20
45	Sön i crëps	Auf den Gipfeln	Auf den Bergen	114	114	aboc sorëdl	viel Sonne	zeitweise Sonne	19
46	da nio	von nichts	bewölkt	106	115	plü tömia	kältere	feuchtere	19
47	raiun dles Alpes	Alpenregion	Alpen	105	116	Variabl y da nio	Variable von nichts	Wechselhaft und bewölkt	19
48	niores	Schneefälle	Wolken	102	117	bel plan	freundlich langsam	allmählich	18
49	niores a gröm	größere Wolken	Quellwolken	97	118	bel plan	gut	allmählich	18
50	minimes	Tiefsttemperaturen	Tiefstwerte	97	119	aboc	meistens	zeitweise	17
51	cresta de confin	dem Kamm	dem Alpenhauptkamm	96	120	naota	noch	zunächst	17
52	Ciadëna zentrala	zentralen Alpenhauptkamm	Alpenhauptkamm	95	121	niores	Schneefelder	Wolken	17
53	de transiziun	Übergangsdruck	Zwischenhoch	90	122	stopa sovënz la	beeinträchtigen oft die	behindern oft die	14
54	ciarü	klar	Hochnebel	85	123	Sorëdl y niores a	Sonnenschein und Schnee in	Sonne und Quellwolken	14
55	y danmisdé	und übermorgen	und am Vormittag	84	124	niores	Jüngeren	Wolken	14
56	niores	Berge	Wolken	83	125	niores a slaier	Wolken zum Anpflanzen	Schleierwolken	12
57	tömia	kühle	feuchte	82	126	Valgamia da sorëdl	Recht sonnig aus	Recht sonnig	12
58	Tres plü instabil	Drei instabiler	Zunehmend unbeständig	77	127	banc de ciarü	Schneebänke	Nebelfelder	12
59	domisdé	heute Morgen	heute Nachmittag	76	128	indlunch	später	überall	12
60	romagn variabl	bleibt variabel	bleibt wechselhaft	76	129	Da nio	Von Schnee	Bewölkt	12
61	Al romagn variabl	Es bleibt variabel	Es bleibt wechselhaft	76	130	Sön la Ciadëna zentrala dles Alpes	Auf der zentralen Alpenhauptkamm der Alpen	Am Alpenhauptkamm	11
62	ploiüdes	Schauern	Regenschauern	76	131	niores a gröm	Wolken in Haufen	Quellwolken	10
63	Mioramënt dl	Erinnerung an die Zeit	Wetterbesserung	73	132	Dantadöt da nio	Dank von nichts	Wolken überwiegen	10
64	nio	Schnee	Wolken	72	133	ciarü alt	klare Höhe	Hochnebel	9
65	y cialt	und Wärme	und Warm	71	134	gnanca na niora	nicht einmal eine Wolke	wolkenlos	9
66	Le tëmp	Die Zeit	Das Wetter	69	135	a gröm	Quellwolken Haufen	Quellwolken	8
67	Sö por munt	Oben auf dem Berg	Auf den Bergen	69	136	N pice mioramënt	Ein kleinerer Fortschritt	Leichte Wetterbesserung	6
68	Sorëdl y niores	und Blumen	und Wolken	66	137	Da sorëdl y da nio	Von Sonne und von Nichts	Sonne und Wolken	6
69	bones condiziuns	guten Bedingungen	gute Verhältnisse	62	138	Na presciun alta temporanea	Ein hoher temporärer Gefängnisaufenthalt	Ein Zwischenhoch	5

Table 3: All 138 fixes applied to the synthesised Ladin–German corpus.