# Improving Word Alignment Using Semi-Supervised Learning

**Zhongtao Miao**[1], **Qiyu Wu**[1*], **Masaaki Nagata**[2], **Yoshimasa Tsuruoka**[1]
[1]The University of Tokyo, Tokyo, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[1]{miao, qiyuw, tsuruoka}@logos.t.u-tokyo.ac.jp
[2]masaaki.nagata@ntt.com

## Abstract

Word alignment plays a crucial role in various natural language processing tasks, such as serving as cross-lingual signals for sentence embedding, reducing hallucination and omission in machine translation, and facilitating the construction of training data for simultaneous speech translation. Current state-of-the-art approaches usually rely on: (1) supervised data and large-scale weakly supervised data constructed from Wikipedia and (2) multilingual Transformer encoder-based models. However, we find that the current state-of-the-art encoder-based method, BinaryAlign, suffers from the issue of insufficient labeled data, and we further improve it with self-training with a small amount of parallel data. In addition, considering the impressive performance of multilingual large language models on many natural language processing tasks, we also explore the possibility of using these decoder-based large language models as word aligners. We observe that although fine-tuning large language models with labeled data produces acceptable results, augmenting the training with pseudo-labeled data further enhances model performance. Based on the findings, we propose a semi-supervised framework to improve the large language model-based word aligners. Experimental results demonstrate that the proposed method with a small amount of parallel data outperforms the current state-of-the-art method on various word alignment datasets.

## 1 Introduction

Word alignment aims to identify correspondences between source and target words in a translation sentence pair, as shown in Figure 1. Although word alignment was initially proposed to enhance statistical machine translation (Brown et al., 1993), advancements in both word alignment and deep learning techniques have broadened its application

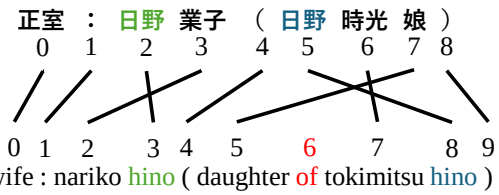


Figure 1: A Japanese-English word alignment example. Note that word alignment not only takes care of the corresponding word(s) within the source and target sentences but also considers the positional information. The English words hino in green and hino in blue are different for the Japanese word 日野, which apprears in the second position (zero-indexed) of the Japanese sentence. The English word of in red does not have the corresponding Japanese word (null word alignment).

to a wide range of tasks in natural language processing (NLP). For example, Chi et al. (2021) use word alignment in cross-lingual language model pretraining. Word alignment is utilized by Miao et al. (2024a) to improve cross-lingual sentence embedding. Wu et al. (2024) leverage word alignment preference data to mitigate hallucination and omission in machine translation models. Word alignment can also be utilized in the simultaneous speech translation domain for data curation (Ouyang et al., 2025b; Fu et al., 2025). Other possible applications include alleviating the over-translation and under-translation problems in machine translation (Tu et al., 2016), XML-structured parallel text segment extraction (Hashimoto et al., 2019) and constrained neural machine translation (Song et al., 2019; Chen et al., 2021).

Representative studies for improving word alignment with supervised methods include SpanAlign (Nagata et al., 2020), WSPAlign (Wu et al., 2023) and BinaryAlign (Latouche et al., 2024). SpanAlign (Nagata et al., 2020) reformulates the word alignment task as a SQuAD-style span prediction question answering task. WSPAlign (Wu et al., 2023) relaxes the requirement of

*Qiyu Wu contributed to this paper when he was a student at The University of Tokyo.



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 19871–19888

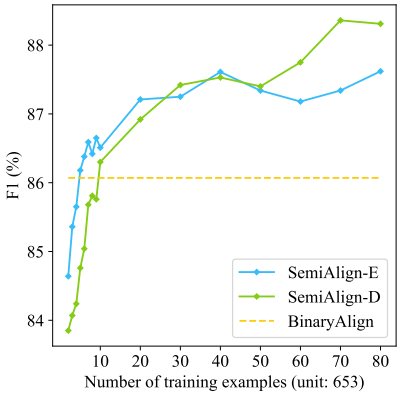


Figure 2: Trend of F1 scores as the amount of unlabeled parallel sentences in the training data increases, using our semi-supervised framework on the test set of the Japanese-English word alignment dataset, KFTT (Neubig, 2011). Note that one unit (653) is the size of the labeled data. The yellow dotted line denotes the F1 score of the current state-of-the-art method, BinaryAlign (Latouche et al., 2024).

manually labeled data in SpanAlign with a *large-scale* weak supervision pretraining dataset (2 million noisy sentence pairs) based on entity links from Wikipedia and contextual word embeddings from multilingual encoder-based language models. BinaryAlign (Latouche et al., 2024) further reformulates the word alignment task as a binary sequence labeling task, achieving the new state-of-the-art performance.

These supervised methods for word alignment are highly dependent on (1) manually labeled datasets (Nagata et al., 2020; Latouche et al., 2024) or large-scale weak word alignment signals extracted from Wikipedia entity links and contextual word embeddings of encoder-based models to relax the strict requirement of labeled data (Wu et al., 2023) and (2) multilingual encoder models (Nagata et al., 2020; Wu et al., 2023; Latouche et al., 2024). This raises the following questions: (1) can the potential of multilingual encoder-based models be fully leveraged by only using the labeled datasets for the current state-of-the-art method, BinaryAlign? If not, is there a way of using a small amount of data to improve it instead of utilizing a large-scale weakly supervised pre-training dataset like WSPAlign? (2) given the success of LLMs across a wide range of NLP tasks, how effective are multilingual LLMs for the word alignment task?

We find that the issue of insufficient labeled data in word alignment prevents both multilingual encoder-based and decoder-based large language models from fully utilizing their advantages, leading to suboptimal performance as shown in Figure 2. Figure 2 presents the trend of F1 scores on the Japanese word alignment dataset KFTT, as we progressively increase the number of parallel sentences used for training our multilingual encoder (SemiAlign-E) and decoder (SemiAlign-D) models.

Based on the aforementioned findings, we propose a semi-supervised framework named *SemiAlign*, which is short for leveraging **Semi**-supervised learning to improve word **Align**ment, designed to alleviate the challenge of limited labeled data in word alignment by utilizing a small amount of unlabeled parallel text. We demonstrate the effectiveness of our method for both multilingual encoder-based and decoder-based language models. The contributions of this paper can be summarized as follows:

- We find that the current state-of-the-art method, BinaryAlign (Latouche et al., 2024), which is based on a multilingual encoder model, suffers from the issue of limited labeled data, leading to suboptimal performance.
- We also observe the issue of limited labeled data when fine-tuning multilingual decoder-based language models as word aligners. We investigate the important factors that affect the word alignment performance of multilingual decoder language models.
- We propose a semi-supervised framework to mitigate the issue of limited labeled data in both scenarios, and the proposed method achieves new state-of-the-art performance on all language pairs that we evaluate with a small amount of unlabeled data.

## 2 Methodology

**Background.** Given a source sentence $\mathbf{x} = [x_1, \ldots, x_n]$ and its target sentence $\mathbf{y} = [y_1, \ldots, y_m]$, where $x_i$ denotes the $i$-th word in the source sentence ($1 \leq i \leq n$) and $y_j$ represents the $j$-th word in the target sentence ($1 \leq j \leq m$), word alignment aims to find a set of source and target word pairs in sentences $\mathbf{x}$ and $\mathbf{y}$:

$$\mathcal{A} = \{(x_i, y_j) : x_i \in \mathbf{x}, y_j \in \mathbf{y}\}, \quad (1)$$

The output space is a set of two-element ordered tuples. $x_i$ and $y_j$ in each pair of $\mathcal{A}$ are the words that

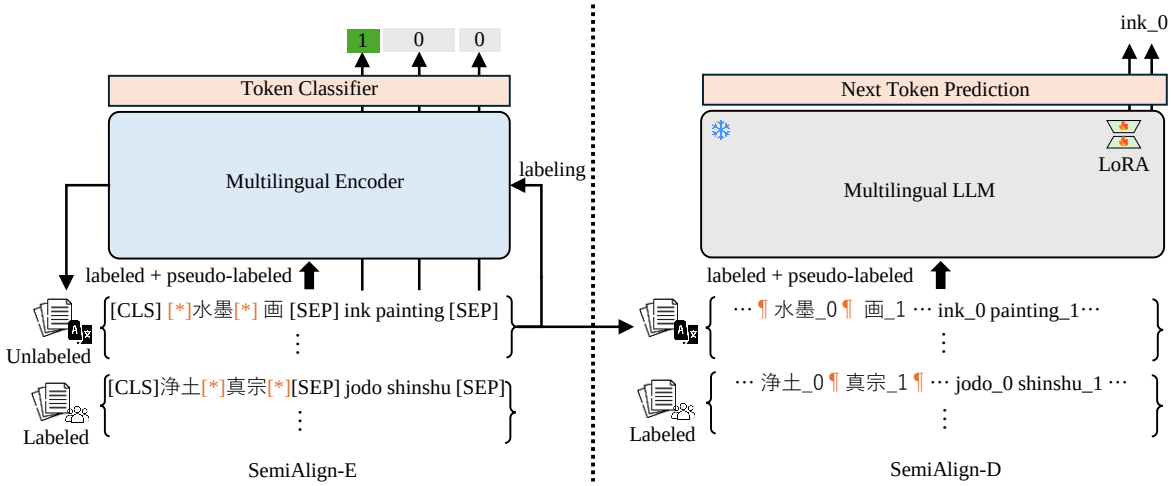


Figure 3: The proposed semi-supervised framework, SemiAlign, illustrated with the Japanese-English training process as an example. The left sub-figure illustrates the process of training multilingual encoder models with self-training. The right sub-figure presents the process of training multilingual decoder models using LoRA with the combination of the pseudo-labeled data predicted by the encoder models and the labeled data. Note that for decoder models, we append positional information to each word to resolve positional ambiguities arising from repeated occurrences of the same word in a sentence. The detailed prompt example for the right sub-figure can be found in Figure 5 (Appendix A.6).

are semantically equivalent to each other within the context of sentences $\mathbf{x}$ and $\mathbf{y}$. The $x_i$ and $y_i$ are typically represented as the positions of words in the source and target sentences because word alignment considers positional information. It is worth noting that in some word alignment datasets, each alignment pair in the set $\mathcal{A}$ is annotated with one of two types of labels: possible alignment ($\mathcal{P}$) and sure alignment ($\mathcal{S}$).

### 2.1 Unlabeled Data Collection and Preparation

To improve word alignment with semi-supervised learning, we first need unlabeled data. Unlike WSPAlign, which uses entity hyperlinks in Wikipedia for Wikipedia entity words and contextual word embeddings for common words as weak word alignment signals, we utilize parallel corpora that do not contain any weak signals. The parallel corpora are denoted as $\mathcal{D}_u$.

**Preprocessing of Unlabeled Data.** Given the raw parallel sentences, we perform some preprocessing steps for further training. The first step is tokenization. For English sentences, we use the Moses tokenizer (Koehn et al., 2007), while for other languages, we employ corresponding language-specific tokenizers. The second step involves lowercasing the parallel sentences if the labeled data is in lowercase. This step is omitted for Chinese and Japanese, as these languages do not distinguish between uppercase and lowercase characters. Beyond the aforementioned steps, we also remove any empty lines in the parallel corpora. As the parallel corpora are sourced from different origins, some may have been partially preprocessed. In such cases, we omit the corresponding preprocessing steps that have been applied.

### 2.2 Multilingual Encoders: Self-Training

For encoder models, we utilize the idea of self-training (Amini et al., 2025) to fully leverage the power of encoder-based language models and mitigate the issue of limited labeled data instead of using weak supervision signals from Wikipedia entity links (Wu et al., 2023). After the preprocessing of unlabeled data, we utilize the current state-of-the-art model, BinaryAlign, to generate the pseudo-labels for unlabeled data $\mathcal{D}_u$. The pseudo-labeled data is denoted as $\mathcal{D}_p$. Then we combine the pseudo-labeled data $\mathcal{D}_p$ and the labeled data $\mathcal{D}_l$ to re-train an encoder model via the binary sequence labeling task of BinaryAlign, as shown in the left sub-figure of Figure 3. After training, we can obtain SemiAlign-E.

### 2.3 Multilingual LLMs: Exploring the Possibility as Word Aligners

In this section, we first evaluate the performance of LLMs that are fine-tuned on labeled data to gain insights into the effectiveness of LLM-based word aligners using standard supervised fine-tuning. Subsequently, we investigate whether incorporating semi-supervised learning can further improve the performance of LLM-based word aligners. Given that generating labels with LLM-based word aligners is relatively computationally intensive, we primarily leverage the pseudo-labeling capabilities of encoder-based methods. Specifically, we employ the trained encoder model, SemiAlign-E, from the previous section to generate pseudo-labels for the unlabeled data.

**Controlling Generation Granularity.** Word alignment can be reformulated as different tasks (Nagata et al., 2020; Latouche et al., 2024). Previous studies (Nagata et al., 2020; Wu et al., 2023) define the task as a SQuAD question answering task via span prediction or a binary sequence labeling task (Latouche et al., 2024). For decoder-based LLMs, it is necessary to reformulate the word alignment task as a generative task while incorporating positional information. Additionally, the generation granularity should be carefully considered when feeding translation pairs into these LLMs. However, the optimal generation granularity for LLM-based word aligners remains underexplored. We investigate two levels of generation granularity: (1) "full mode", where the model outputs all aligned word pairs with their positions given the source and target sentence in a single pass; and (2) "marker mode", where a specific word in the source sentence is marked with special tokens, and the model outputs its corresponding aligned words with positions in the target sentence. The special tokens are represented by the orange symbols in the right panel of Figure 3. The prompt formats for "full mode" and "marker mode" can be found in Appendix A.6.

We use the cross-entropy loss for next token prediction to train our decoder models. The training data is the combination of pseudo-labeled data and labeled data. We mask the instruction prompt part when computing the loss during the training following self-instruct (Wang et al., 2023). To augment the training data, we incorporate both source-to-target and target-to-source alignment directions.

**Symmetric Alignment for LLMs.** For the marker mode generation, we mark one word in the source sentence and let the model predict its corresponding words and positions in the target sentence. This prediction can also be performed in the other direction. Following SpanAlign (Nagata et al., 2020) and WSPAlign (Wu et al., 2023), we perform the symmetric alignment during evaluation or prediction. Specifically, we obtain predictions from both alignment directions and combine them via union operation, preserving all predictions from both directions.

## 3 Experiments

### 3.1 Datasets

We mainly evaluate the performance of the proposed method on four language pairs: Japanese-English (ja-en), German-English (de-en), Romanian-English (ro-en) and Chinese-English (zh-en). We use ISO 639-1 language codes[1] to denote languages in the following tables. Specifically, "ja" denotes Japanese. "zh" denotes Chinese. "ro" denotes Romanian. "de" represents German. "en" corresponds to English. We also test our method on some low-resource languages including Swedish (sv), Finnish (fi), and Hebrew (he).

**Labeled Data.** For Japanese-English (ja-en) word alignment, we use The Kyoto Free Translation Task (KFTT) word alignment dataset (Neubig, 2011). In this dataset, we adopt the same data splits as those used in SpanAlign (Nagata et al., 2020), WSPAlign (Wu et al., 2023) and BinaryAlign (Latouche et al., 2024) with the script available at the repository of SpanAlign[2]. The Romanian-English word alignment dataset comes from Mihalcea and Pedersen (2003). The German-English word alignment dataset is provided by Vilar et al. (2006). The Chinese-English word alignment dataset is obtained from the TsinghuaAligner website[3]. We use the v1 version of the dataset following BinaryAlign. Note that we use the same splitting approach as WSPAlign following BinaryAlign in the above datasets. Moreover, we utilize the Finnish-Greek (fi-el) and the Finnish-Hebrew (fi-he) word alignment datasets from Imani et al. (2021) and the

[1] https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes

[2] https://github.com/nttcslab-nlp/word_align

[3] https://nlp.csai.tsinghua.edu.cn/~ly/systems/TsinghuaAligner/TsinghuaAligner.html

| Lang | Method | Precision (%) | Recall (%) | F1 (%) ↑ | AER (%) ↓ |
|---|---|---|---|---|---|
| ja-en | SpanAlign (Nagata et al., 2020) | 77.3 | 78.0 | 77.6 | 22.4 |
| | WSPAlign (Wu et al., 2023) | 81.6 | 85.9 | 83.7 | 16.3 |
| | BinaryAlign* (Latouche et al., 2024) | 87.74 | 84.46 | 86.07 | 13.93 |
| | Llama3.1 Inst + SFT | 78.65 | 89.76 | 83.84 | 16.16 |
| | SemiAlign-E (ours) | 88.34 | 86.91 | 87.62 | 12.38 |
| | SemiAlign-D (ours) | 88.09 | 88.63 | **88.36** | **11.64** |
| de-en | SpanAlign (Nagata et al., 2020) | 89.9 | 81.7 | 85.6 | 14.4 |
| | WSPAlign (Wu et al., 2023) | 90.7 | 87.1 | 88.9 | 11.1 |
| | BinaryAlign* (Latouche et al., 2024) | 93.79 | 90.73 | 92.23 | 7.74 |
| | Llama3.1 Inst + SFT | 85.46 | 90.98 | 88.14 | 11.93 |
| | SemiAlign-E (ours) | 94.55 | 90.93 | **92.70** | **7.27** |
| | SemiAlign-D (ours) | 93.97 | 90.69 | 92.31 | 7.66 |
| ro-en | SpanAlign (Nagata et al., 2020) | 90.4 | 85.3 | 86.7 | 12.2 |
| | WSPAlign (Wu et al., 2023) | 92.0 | 90.9 | 91.4 | 8.6 |
| | BinaryAlign* (Latouche et al., 2024) | 92.67 | 92.51 | 92.59 | 7.41 |
| | Aya Expanse + SFT | 80.37 | 87.72 | 83.89 | 16.11 |
| | SemiAlign-E (ours) | 93.55 | 92.68 | **93.11** | **6.89** |
| | SemiAlign-D (ours) | 94.65 | 89.52 | 92.02 | 7.98 |
| zh-en | SpanAlign (Nagata et al., 2020) | - | - | - | 8.9 |
| | WSPAlign (Wu et al., 2023) | - | - | - | 7.6 |
| | BinaryAlign* (Latouche et al., 2024) | 95.63 | 94.13 | 94.87 | 5.12 |
| | Llama3.1 Inst + SFT | 92.60 | 96.00 | 94.27 | 5.73 |
| | SemiAlign-E (ours) | 95.57 | 95.14 | 95.35 | 4.65 |
| | SemiAlign-D (ours) | 96.18 | 95.13 | **95.65** | **4.35** |

Table 1: Results of different methods on word alignment datasets. SemiAlign-E denotes the multilingual encoder models that are trained with the proposed method. SemiAlign-D indicates the multilingual large language models that are trained with our framework. "SFT" indicates the results of supervised fine-tuning using labeled word alignment datasets. "BinaryAlign*" denotes the best reproduced results of BinaryAlign with precision, recall, F1 and AER shown to give a detailed comparison. We highlight in bold the best performance of F1 and AER scores because higher F1 and lower AER scores indicate better performance.

English-Swedish (en-sv) word alignment dataset from Holmqvist and Ahrenberg (2011) to evaluate the performance of various models on low-resource languages. The number of training, validation and test examples in the labeled data can be found in Table 10.

**Unlabeled Data.** For Japanese-English, we use a subset of the KFTT translation data (Neubig, 2011). For Romanian-English, we use the training parallel corpus[4] from Vilar et al. (2006). For the German-English pair, we use a subset of News Commentary v18.1 (Kocmi et al., 2023) as the unlabeled data. For Chinese-English pair, we use the News Commentary v16. For English-Swedish, Finnish-Greek and Finnish-Hebrew pairs, we utilize subsets of CCMatrix v1.0 (Schwenk et al., 2021; Fan et al., 2021; Tiedemann, 2012). The max number of unlabeled data that we use is shown in Table 11. More details about the unlabeled data can be found in Appendix A.2

### 3.2 Evaluation Metrics

We evaluate the word alignment quality using Precision (P), Recall (R), F1 and Alignment Error Rate (AER) (Och and Ney, 2003) for all experiments following previous works (Nagata et al., 2020; Wu et al., 2023). The details of evaluation metrics can be found in Appendix A.5.

Note that BinaryAlign (Latouche et al., 2024) only presents the AER scores for all language pairs. We reimplement BinaryAlign using BinaryAlign repository[5] and present the best reproduced precision, recall, F1 and AER scores for a fair and detailed comparison.

### 3.3 Models

To evaluate the effectiveness of our method, we consider two types of multilingual language models: encoder-based and decoder-based language models.

**Multilingual Encoders.** We utilize mDeBERTa-v3-base[6] (He et al., 2023) as our base model following BinaryAlign. For the reproduced BinaryAlign

[4] http://web.eecs.umich.edu/~mihalcea/wpt/data.protected/Romanian-English.training.tar.gz

[5] https://github.com/ubisoft/ubisoft-laforge-BinaryAlignWordAlignementasBinarySequenceLabeling/

[6] https://huggingface.co/microsoft/mdeberta-v3-base

experiments, we also use mDeBERTa-v3-base as the base model for a fair comparison. All hyperparameters in the reproduced BinaryAlign experiments are the same as BinaryAlign. We present the best reproduced results on all language pairs. For the experiments that use pseudo-labeled data, the total batch size is 32. The number of epoch is 1. For other hyperparameters, we follow the setting in BinaryAlign which can be found in its GitHub repository.

**Multilingual LLMs.** For decoder-based LLMs, we mainly use Llama-3.1-8B-Instruct (Dubey et al., 2024) as our base model. The only exception is the Romanian English pair because we find that the best result of the Romanian-English pair is based on the aya-expanse-8b (Dang et al., 2024). The total batch size is 32. The initial learning rate is 2e-5. The scheduler type is "cosine". We utilized LoRA for parameter-efficient training. For the experiments that leverage pseudo-label data, the number of epochs is 1. We use bf16 mixed precision training. The optimizer is "adamw_torch". The weight decay is set to 0. Additional training details and LoRA settings can be found in Appendix A.3 and A.4.

### 3.4 Baselines

We mainly compare our method with three approaches: SpanAlign, WSPAlign and BinaryAlign. SpanAlign is a supervised word alignment training method that reformulates the word alignment task as a question answering task (Rajpurkar et al., 2018) via span prediction. WSPAlign focuses on utilizing large-scale weak signals from entity hyperlinks in Wikipedia articles for word alignment pre-training to improve SpanAlign. BinaryAlign reformulates the word alignment task as a binary sequence labeling task achieving a good balance between outputting all word alignments at once (all-at-once) and predicting one target word for each source word (one-by-one way) in one pass. For multilingual decoder-based LLMs, we present the results of the models that are supervised fine-tuned on labeled data for reference (SFT row) in Table 1.

### 3.5 Low-Resource Settings

**Zero-Shot Setting.** In the zero-shot setting, the training sets of the word alignment datasets for our target language pairs are not available. We assess the performance of our models that are trained on a different language pair on the test set of the target word alignment dataset directly. We use the models that are trained on the word alignment datasets of high-resource language pairs (ja-en, ro-en, de-en and zh-en) and evaluate their performance on three low-resource languages, that is, English-Swedish (en-sv), Finnish-Greek (fi-el), and Finnish-Hebrew (fi-he).

**Few-Shot Setting.** In this setting, we train our models with few-shot examples for a language pair and evaluate the models in the same language pair. Specifically, we use 32 examples from the training sets of word alignment datasets to serve as supervision signals. For SemiAlign-D, we use the same number of pseudo-labeled examples as the labeled examples (32), considering the training cost. For SemiAlign-E, we evaluate two different settings for pseudo-labeled data: one using 32 pseudo-labeled examples and the other using the pseudo-labeled examples that yield the best results in the fully supervised setting. We present the superior results from these two configurations. For the pseudo-labelers, we choose the BinaryAlign models that are trained with 32 examples for each language pair. These models also serve as our baseline methods for comparison. We evaluate different approaches in this setting with seven language pairs (ja-en, ro-en, de-en, zh-en, en-sv, fi-el and fi-he).

## 4 Results and Analysis

The main results are presented in Table 1, demonstrating that the proposed method achieves new state-of-the-art performance. The Japanese-English word alignment dataset shows the most significant improvement. For the encoder-based model, the AER score of SemiAlign-E improves from 13.93 to 12.38 compared to BinaryAlign. For decoder-based LLMs, the AER score of SemiAlign-D improves from 16.16 to 11.64 compared with its SFT baseline. The average F1 and AER scores of SemiAlign-E and SemiAlign-D across the four datasets differ by about 0.1 and better than BinaryAlign. On average, the improvement in AER scores across the four datasets shows that SemiAlign-D achieves a greater performance gain over the SFT baseline compared to SemiAlign-E's improvement over BinaryAlign. This indicates that the size of the training data has a relatively larger impact on multilingual large language models.

**A Small Amount of Unlabeled Data Boosts the Performance of BinaryAlign.** We re-train Bi-

| Lang | Method | P | R | F1 | AER |
|---|---|---|---|---|---|
| en-sv | BinaryAlign (ja-en) | 94.54 | 97.28 | 95.89 | 4.21 |
| | SemiAlign-D (ja-en) | 91.42 | 92.87 | 92.14 | 7.91 |
| | SemiAlign-E (ja-en) | 94.77 | 97.63 | **96.18** | **3.92** |
| | BinaryAlign (ro-en) | 91.55 | 97.25 | 94.31 | 5.95 |
| | SemiAlign-D (ro-en) | 91.04 | 92.49 | 91.76 | 8.30 |
| | SemiAlign-E (ro-en) | 92.38 | 97.43 | **94.84** | **5.40** |
| | BinaryAlign (de-en) | 95.73 | 96.86 | 96.29 | 3.75 |
| | SemiAlign-D (de-en) | 95.37 | 94.61 | 95.00 | 4.98 |
| | SemiAlign-E (de-en) | 96.60 | 96.86 | **96.73** | **3.28** |
| | BinaryAlgin (zh-en) | 98.24 | 93.5 | 95.81 | 4.11 |
| | SemiAlign-D (zh-en) | 97.44 | 93.92 | 95.65 | 4.24 |
| | SemiAlign-E (zh-en) | 98.30 | 93.95 | **96.08** | **3.85** |
| fi-el | BinaryAlign (ja-en) | 83.09 | 76.69 | 79.76 | 20.24 |
| | SemiAlign-D (ja-en) | 82.07 | 76.18 | 79.02 | 20.98 |
| | SemiAlign-E (ja-en) | 83.03 | 78.53 | **80.72** | **19.28** |
| | BinaryAlign (ro-en) | 80.87 | 84.20 | 82.50 | 17.50 |
| | SemiAlign-D (ro-en) | 82.74 | 73.75 | 77.99 | 22.01 |
| | SemiAlign-E (ro-en) | 83.76 | 83.62 | **83.69** | **16.31** |
| | BinaryAlign (de-en) | 86.66 | 70.32 | 77.64 | 22.36 |
| | SemiAlign-D (de-en) | 87.63 | 77.27 | **82.13** | **17.87** |
| | SemiAlign-E (de-en) | 87.09 | 71.89 | 78.76 | 21.24 |
| | BinaryAlign (zh-en) | 88.87 | 68.47 | 77.35 | 22.65 |
| | SemiAlign-D (zh-en) | 86.15 | 77.49 | **81.59** | **18.41** |
| | SemiAlign-E (zh-en) | 87.84 | 70.03 | 77.93 | 22.07 |
| fi-he | BinaryAlign (ja-en) | 70.54 | 63.49 | 66.83 | 33.17 |
| | SemiAlign-D (ja-en) | 71.06 | 62.03 | 66.24 | 33.76 |
| | SemiAlign-E (ja-en) | 72.96 | 62.86 | **67.54** | **32.46** |
| | BinaryAlign (ro-en) | 70.83 | 57.08 | 63.22 | 36.78 |
| | SemiAlign-D (ro-en) | 77.90 | 58.96 | **67.12** | **32.88** |
| | SemiAlign-E (ro-en) | 79.39 | 55.35 | 65.23 | 34.77 |
| | BinaryAlign (de-en) | 82.58 | 40.07 | 53.96 | 46.04 |
| | SemiAlign-D (de-en) | 84.52 | 58.86 | **69.39** | **30.61** |
| | SemiAlign-E (de-en) | 86.70 | 42.48 | 57.02 | 42.98 |
| | BinaryAlign (zh-en) | 84.87 | 54.04 | **66.04** | **33.96** |
| | SemiAlign-D (zh-en) | 62.16 | 40.20 | 48.82 | 51.18 |
| | SemiAlign-E (zh-en) | 81.34 | 55.43 | 65.93 | 34.07 |

Table 2: Zero-shot evaluation results. The first column denotes the test language pair. The language pairs in parentheses in the Method column indicate the training language pair.

naryAlign using self-training with one iteration. All hyperparameter settings are the same as BinaryAlign except that the number of epochs is set to 1 and the batch size is changed from 8 to 32. We can observe that the self-training with the small-scale corpora boost the performance of BinaryAlign on all language pairs in Table 1. The most obvious one is Japanese-English pair and the self-training improves the AER score by 1.6.

**Utilizing Labeled Data Is Not Enough for Multilingual LLM-Based Word Aligners.** For a better comparison and investigation of factors that affect the word alignment performance of decoder-based language models, we mainly fine-tune the Llama-3.1-8B-Instruct for all languages with the labeled data as the baseline. For Romanian-English, we use the aya-expanse-8b model because we find that its performance on this language pair is better. For the SFT version of these large language models, the number of epochs is 5. The other hyperparameters are the same as the main experiments. In Table 1, we find that the performance of using the labeled data is not competitive compared with BinaryAlign. We also observe that the proposed framework with a small amount of parallel data improves the decoder-based word aligners by 1.4 on Chinese-English, 4.5 on Japanese-English, 4.3 on German-English, 6.68 on Romanian-English respectively. This indicates that the multilingual large language model-based word aligners are leveraging the pseudo-labeled data efficiently.

**Zero-Shot Setting.** The results of zero-shot setting are shown in Table 2. For the models that are trained on Japanese-English data and evaluated on other language pairs, SemiAlign-E achieves the best performance. For the models that are trained on Romanian-English data, SemiAlign-E is a good choice in most cases. Overall, SemiAlign maintains its generalization performance on low-resource languages and, in most cases, SemiAlign-E even outperforms BinaryAlign. The reason may be that the base model of SemiAlign-E has been pre-trained on data from a large number of languages. In contrast, the base models of SemiAlign-D, such as Llama-3.1-8B-Instruct and aya-expanse-8b, are typically pretrained on a relatively smaller number of languages during the pre-training phase.

| Lang | Method | P | R | F1 | AER |
|---|---|---|---|---|---|
| ja-en | BinaryAlign | 81.86 | 72.33 | 76.8 | 23.2 |
| | SemiAlign-D | 68.44 | 78.04 | 72.93 | 27.07 |
| | SemiAlign-E | 84.10 | 73.39 | **78.38** | **21.62** |
| ro-en | BinaryAlign | 90.65 | 90.92 | 90.78 | 9.22 |
| | SemiAlign-D | 76.75 | 85.86 | 81.05 | 18.95 |
| | SemiAlign-E | 91.80 | 91.18 | **91.49** | **8.51** |
| de-en | BinaryAlign | 93.07 | 90.49 | 91.76 | 8.21 |
| | SemiAlign-D | 83.10 | 87.45 | 85.22 | 14.83 |
| | SemiAlign-E | 93.34 | 90.90 | **92.10** | **7.87** |
| zh-en | BinaryAlign | 94.73 | 91.40 | 93.04 | 6.96 |
| | SemiAlign-D | 89.67 | 90.24 | 89.96 | 10.04 |
| | SemiAlign-E | 95.07 | 91.39 | **93.19** | **6.81** |
| en-sv | BinaryAlign | 95.96 | 97.46 | 96.70 | 3.34 |
| | SemiAlign-D | 87.17 | 92.99 | 89.99 | 10.18 |
| | SemiAlign-E | 96.73 | 97.22 | **96.97** | **3.04** |
| fi-el | BinaryAlign | 91.06 | 86.22 | 88.58 | 11.42 |
| | SemiAlign-D | 82.79 | 87.69 | 85.17 | 14.83 |
| | SemiAlign-E | 91.48 | 87.79 | **89.60** | **10.40** |
| fi-he | BinaryAlign | 86.03 | 76.21 | 80.82 | 19.18 |
| | SemiAlign-D | 78.00 | 88.54 | **82.94** | **17.06** |
| | SemiAlign-E | 86.85 | 79.17 | 82.83 | 17.17 |

Table 3: Few-shot evaluation results. The first column denotes the training and test language pair. The number of available labeled word alignment examples is 32.

**Few-Shot Setting.** We present the results of different methods in the few-shot setting in Table 3. In this setting, we only use 32 labeled examples to train our base models, and generate pseudo-labeled examples for unlabeled data using the

| Lang | Method | # Parameters | # Examples | Inference Time |
|---|---|---|---|---|
| ja-en | BinaryAlign/SemiAlign-E | 276M | 357 | 0h:00m:43s |
| | SemiAlign-D | 8B | | 2h:03m:15s |
| ro-en | BinaryAlign/SemiAlign-E | 276M | 98 | 0h:00m:23s |
| | SemiAlign-D* | 8B | | 1h:01m:12s |
| de-en | BinaryAlign/SemiAlign-E | 276M | 208 | 0h:00m:13s |
| | SemiAlign-D | 8B | | 1h:04m:11s |
| zh-en | BinaryAign/SemiAlign-E | 276M | 450 | 0h:01m:02s |
| | SemiAlign-D | 8B | | 3h:12m:48s |

Table 4: Actual inference time on the test sets of word alignment datasets. SemiAlign-D* in the ro-en row indicates that we use aya-expanse-8b as the base model rather than llama-3.1-8b-instruct model.

model trained with 32 labeled examples. After that, we re-train our base models with our semi-supervised framework. We find that SemiAlign-E usually achieves better performance in this setting compared with BinaryAlign and SemiAlign-D except the fi-he pair. We observe that, in fi-he pair, SemiAlign-D that is trained with 32 pseudo-labeled examples achieves the best performance.

**Error Analysis.** We conduct an error analysis to explore scenarios in which the proposed method fails. Specifically, we collect all incorrect predictions in the test sets of the Japanese-English, German-English, Romanian-English, and Chinese-English word alignment datasets at the word level. We categorize the incorrect predictions into three types: (1) refusal errors, where the model produces no target word predictions despite the existence of valid alignments; (2) single misalignment error, where one incorrect target word is predicted; and (3) multiple misalignment errors, where several incorrect target words are generated for a single source word. We also present an analysis of incorrect predictions obtained by examining each source words that have gold alignments.

The statistics of incorrect predictions of SemiAlign-D and SemiAlign-E by iterating over each source word and the source words that have gold alignments are shown in Table 5, 6, 7, and 8 correspondingly.

| Lang | # refusal (%) | # single misalignment (%) | # multiple misalignments (%) |
|---|---|---|---|
| ja-en | 20.40 | 46.20 | 33.40 |
| de-en (w. p) | 45.40 | 36.35 | 18.24 |
| de-en (w/o. p) | 27.50 | 52.92 | 19.58 |
| ro-en | 42.77 | 41.33 | 15.90 |
| zh-en (w. p) | 39.61 | 40.02 | 20.36 |
| zh-en (w/o. p) | 27.48 | 51.40 | 21.12 |

Table 5: Distribution of prediction error types for SemiAlign-D, obtained by iterating over each source word of the test sets of different languages. "(w. p)" indicates that we consider possible alignments while "(w/o. p)" denotes that we do not consider them during the analysis.

**Efficiency Analysis.** Table 4 presents the actual inference time on the test sets of word alignment datasets, model sizes, and number of examples. SemiAlign-E has the same efficiency as BinaryAlign because they use the same base encoder models and their inference processes are identical.

For SemiAlign-D, which uses LLMs as the base model, although it achieves the highest scores on ja-en and zh-en as shown in Table 1, SemiAlign-D has the following disadvantages: slower inference speed and larger model size. Therefore, for pipelines or tasks that are sensitive to latency, it is preferable to use a more lightweight encoder model, SemiAlign-E.

| Lang | # refusal (%) | # single misalignment (%) | # multiple misalignments (%) |
|---|---|---|---|
| ja-en | 27.23 | 32.16 | 40.61 |
| de-en (w. p) | 50.77 | 29.45 | 19.79 |
| de-en (w/o. p) | 35.10 | 43.74 | 21.16 |
| ro-en | 48.84 | 34.98 | 16.17 |
| zh-en (w. p) | 54.53 | 19.25 | 26.22 |
| zh-en (w/o. p) | 38.44 | 35.14 | 26.42 |

Table 6: Distribution of prediction error types for SemiAlign-D by iterating over the source words that have gold alignments of the test sets. "(w. p)" indicates that we consider possible alignments while "(w/o. p)" denotes that we do not consider them during the analysis.

| Lang | # refusal (%) | # single misalignment (%) | # multiple misalignments (%) |
|---|---|---|---|
| ja-en | 21.24 | 53.24 | 25.52 |
| de-en (w. p) | 44.84 | 40.11 | 15.04 |
| de-en (w/o. p) | 25.95 | 57.79 | 16.26 |
| ro-en | 18.84 | 51.09 | 30.07 |
| zhen (w. p) | 33.88 | 44.98 | 21.14 |
| zhen (w/o. p) | 22.86 | 55.58 | 21.57 |

Table 7: Distribution of prediction error types for SemiAlign-E by iterating over each source word of the test sets. "(w. p)" indicates that we consider possible alignments while "(w/o. p)" denotes that we do not consider them during the analysis.

**Effect of the generation granularity.** Decoder-based large language models treat the word alignment task as a generative task. Before performing the prediction, we must determine the generation granularity first.

To determine the optimal level of generation granularity for the decoder-based language models, we evaluate the performance of the models across different granularity levels using the labeled data. The results of using different levels

| Lang | # refusal (%) | # single misalignment (%) | # multiple misalignments (%) |
|---|---|---|---|
| ja-en | 27.97 | 41.60 | 30.43 |
| de-en (w. p) | 50.40 | 33.33 | 16.26 |
| de-en (w/o. p) | 33.56 | 48.71 | 17.73 |
| ro-en | 22.91 | 44.05 | 33.04 |
| zhen (w. p) | 47.95 | 23.22 | 28.83 |
| zhen (w/o. p) | 33.03 | 37.84 | 29.13 |

Table 8: Distribution of prediction error types for SemiAlign-E by iterating over the source words that have gold alignments. "(w. p)" indicates that we consider possible alignments while "(w/o. p)" denotes that we do not consider them during the analysis.

of generation granularity are shown in Table 9. As shown in Table 9, we find that the performance of "marker mode" demonstrates significantly superior performance. This indicates that LLM-based word aligners perform well at the word level, outputting the corresponding words given one marked word. The word alignment performance deteriorates when LLM-based word aligners generate all aligned words for a sentence in a single pass.

| Generation Mode | Ja-En | De-En | Ro-En | Zh-En |
|---|---|---|---|---|
| Full Mode | 26.60 | 17.33 | 23.64 | 9.95 |
| Marker Mode | **16.16** | **11.93** | **14.66** | **5.73** |

Table 9: AER scores of different generation granularities using labeled data.

More ablation studies and experimental results are available in Appendix B.

## 5 Related Work

**Word alignment.** The research community has proposed various methods to improve word alignment based on the multilingual Transformer encoder models (Devlin et al., 2019; Conneau et al., 2020) . There are two categories of research that focus on improving word alignment performance on multilingual encoder-based language models (Latouche et al., 2024). One category consists of methods based on contextual word embeddings extracted from encoder models (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Wang et al., 2022a). For example, SimAlign (Jalili Sabet et al., 2020) proposes three new alignment methods based on contextual word embeddings. The other category involves supervised learning for word alignment. SpanAlign (Nagata et al., 2020) redefines the word alignment task as a span prediction-based question answering task. WSPAlign (Wu et al., 2023) proposes large-scale weakly pre-training for word alignment, using entity links and contextual word embeddings for entity and common words, due to the different representation level of them (Schick and Schütze, 2020; Wu et al., 2021). BinaryAlign (Latouche et al., 2024) reformulates the word alignment task as a binary sequence labeling task, achieving the new state-of-the-art performance. Both of the above categories are based on the multilingual encoder language models. However, considering that autoregressive decoder-based LLMs have achieved impressive performance on many NLP tasks, such as coding, reasoning and machine translation (Shojaee et al., 2023; Wu et al., 2024; Dubey et al., 2024; Miao et al., 2024b; Dang et al., 2024), we also explore the possibility of using multilingual LLMs as word aligners, an avenue that remains relatively unexplored by the research community.

**Semi-Supervised Learning.** Semi-supervised learning aims to improve various machine learning systems with labeled data and unlabeled data (Yang et al., 2023). Recently, researchers have successfully applied semi-supervised learning to various emergent domains, such as sentence semantics (Wu et al., 2022; Zhao et al., 2024), multimodal domains (Wang et al., 2022b; Deng et al., 2024; Ouyang et al., 2025a; Hoyer et al., 2025), aspect sentiment quad prediction (Zhang et al., 2024), improving reasoning ability of large language models (Wang et al., 2024) and reward modeling (Li et al., 2024; He et al., 2024). In this work, we investigate the application of semi-supervised learning on word alignment through self-training and pseudo-labeling.

## 6 Conclusion

In this paper, we find that the current state-of-the-art method, BinaryAlign, can be further improved with semi-supervised learning with a small amount of unlabeled data. We also explore the possibility of utilizing multilingual LLMs as word aligners. We observe that fine-tuning multilingual LLMs using the labeled data is insufficient, and not comparable to the current state-of-the-art BinaryAlign. Therefore, we also utilize the pseudo-labeled data from the encoder-based word alignment models to improve multilingual LLM-based word aligners in our proposed framework. Both multilingual encoder models and LLMs that are trained with the proposed method achieve competitive performance, outperforming the current state-of-the-art method on average. The extensive experimental results demonstrate the effectiveness of the proposed method.

## Limitations

We explore the possibility of using multilingual LLMs as word aligners and propose SemiAlign-D. However, compared to SemiAlign-E, the primary limitation of SemiAlign-D lies in their high computational requirements, while their performance in word alignment is not significantly superior to that of SemiAlign-E. In order to fully leverage the potential of LLMs, we need to explore better ways to do word alignment using LLMs in the future. Another interesting future direction is how to mitigate confirmation bias. Confirmation bias is a common dilemma for semi-supervised learning. All pseudo-labelers cannot avoid introducing confirmation bias. Figure 2 in the paper shows the performance trend of our method when the number of pseudo-labeled data increases. From the figure, we find that the impact of confirmation bias is relatively small on the performance of our method. The reason for the minimal impact of confirmation bias in our method might be that our models are trained using a single iteration within the semi-supervised framework. This single iteration minimizes the accumulation of errors in the predicted pseudo-labels during training.

## Ethics Statement

This paper aims to improve the word alignment performance of multilingual pre-trained language models. All datasets that are used in this paper are publicly available without copyright issues. We use mDeBERTa-v3-base, Llama-3.1-8B-Instruct and aya-expanse-8b models in our experiments adhering to the respective model licenses. Note that certain models built upon large language models may generate content irrelevant to word alignment if their prompts are not properly configured.

## References


Massih-Reza Amini, Vasilii Feofanov, Loïc Pauletto, Liès Hadjadj, Émilie Devijver, and Yury Maximov. 2025. Self-training: A survey. *Neurocomputing*, 616:128904.

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

Guanhua Chen, Yun Chen, and Victor O.K. Li. 2021. Lexically constrained neural machine translation with explicit alignment guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12630–12638.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3418–3430, Online. 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.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, Quanquan Gu, James Zou, Kai-Wei Chang, and Wei Wang. 2024. Enhancing large vision language models with self-training on image comprehension. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave,

Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Biao Fu, Donglei Yu, Minpeng Liao, Chengxi Li, Yidong Chen, Kai Fan, and Xiaodong Shi. 2025. Efficient and adaptive simultaneous speech translation with fully unidirectional architecture. *arXiv preprint arXiv:2504.11809*.

Kazuma Hashimoto, Raffaella Buschiazzo, James Bradbury, Teresa Marshall, Richard Socher, and Caiming Xiong. 2019. A high-quality multilingual dataset for structured documentation translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 116–127, Florence, Italy. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Yifei He, Haoxiang Wang, Ziyan Jiang, Alexandros Papangelis, and Han Zhao. 2024. Semi-supervised reward modeling via iterative self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7365–7377, Miami, Florida, USA. Association for Computational Linguistics.

Maria Holmqvist and Lars Ahrenberg. 2011. A gold standard for English-Swedish word alignment. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 106–113, Riga, Latvia. Northern European Association for Language Technology (NEALT).

Lukas Hoyer, David Joseph Tan, Muhammad Ferjad Naeem, Luc Van Gool, and Federico Tombari. 2025. Semivl: Semi-supervised semantic segmentation with vision-language guidance. In *Computer Vision – ECCV 2024*, pages 257–275, Cham. Springer Nature Switzerland.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ayyoob Imani, Masoud Jalili Sabet, Lutfi Kerem Senel, Philipp Dufter, François Yvon, and Hinrich Schütze. 2021. Graph algorithms for multiparallel word alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8457–8469, Online and Punta Cana, Dominican Republic. 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.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

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

Siyuan Li, Weiyang Jin, Zedong Wang, Fang Wu, Zicheng Liu, Cheng Tan, and Stan Z. Li. 2024. Semireward: A general reward model for semi-supervised learning. In *The Twelfth International Conference on Learning Representations*.

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

Zhongtao Miao, Kaiyan Zhao, and Yoshimasa Tsuruoka. 2024b. Improving arithmetic reasoning ability of large language models through relation tuples, verification and dynamic feedback. *arXiv preprint arXiv:2406.17873*.

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

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.

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

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Liangyang Ouyang, Ruicong Liu, Yifei Huang, Ryosuke Furuta, and Yoichi Sato. 2025a. Actionvos: Actions as prompts for video object segmentation. In *Computer Vision – ECCV 2024*, pages 216–235, Cham. Springer Nature Switzerland.

Siqi Ouyang, Xi Xu, and Lei Li. 2025b. Infinisst: Simultaneous translation of unbounded speech with large language model. *arXiv preprint arXiv:2503.02969*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Parshin Shojaee, Aneesh Jain, Sindhu Tipirneni, and Chandan K. Reddy. 2023. Execution-based code generation using deep reinforcement learning. *Transactions on Machine Learning Research*.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

David Vilar, Maja Popovic, and Hermann Ney. 2006. AER: do we need to "improve" our alignments? In *Proceedings of the Third International Workshop on Spoken Language Translation: Papers*, Kyoto, Japan.

Tianduo Wang, Shichen Li, and Wei Lu. 2024. Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928, Bangkok, Thailand. Association for Computational Linguistics.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022b. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

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

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022. PCL: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12052–12066, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu. 2021. Taking notes on the fly helps language pre-training. In *International Conference on Learning Representations*.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2023. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954.

Yice Zhang, Jie Zeng, Weiming Hu, Ziyi Wang, Shiwei Chen, and Ruifeng Xu. 2024. Self-training with pseudo-label scorer for aspect sentiment quad prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11862–11875, Bangkok, Thailand. Association for Computational Linguistics.

Kaiyan Zhao, Qiyu Wu, Xin-Qiang Cai, and Yoshimasa Tsuruoka. 2024. Leveraging multi-lingual positive instances in contrastive learning to improve sentence embedding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 976–991, St. Julian's, Malta. Association for Computational Linguistics.

# A Additional Experimental Details

## A.1 Word Alignment Dataset Statistics

Table 10 shows the number of training, validation and test sets of word alignment datasets that we use covering different levels of supervision.

| Lang | # Train | # Val | # Test |
|---|---|---|---|
| fully supervised | | | |
| ja-en | 653 | 225 | 357 |
| de-en | 300 | - | 208 |
| ro-en | 150 | - | 98 |
| zh-en | 450 | - | 450 |
| few shot | | | |
| ja-en | 32 | 225 | 357 |
| de-en | 32 | - | 208 |
| ro-en | 32 | - | 98 |
| zh-en | 32 | - | 450 |
| en-sv | 32 | - | 192 |
| fi-el | 32 | - | 791 |
| fi-he | 32 | - | 200 |
| zero shot | | | |
| en-sv | - | - | 192 |
| fi-el | - | - | 791 |
| fi-he | - | - | 200 |

Table 10: Number of training, validation and test examples in different supervision settings.

## A.2 Details of Parallel Data

For Japanese-English, we use a subset of the KFTT translation data (Neubig, 2011). Specifically, we use a subset of the cleaned and tokenized version of the training set of the translation data. The original training set contains about 330K sentences. For the tokenized English sentences in the training set, we further preprocessed the text by lowercasing all English tokens.

For Romanian-English, we use the training parallel corpus[7] from Vilar et al. (2006). The parallel corpus comes from three sources, Orwell's 1984, the Romanian Constitution and parallel texts collected from the Web. We remove empty lines and use a subset of the parallel sentences.

For the German-English pair, we use a subset of News Commentary v18.1 (Kocmi et al., 2023) as the unlabeled data. The only preprocessing step before using these parallel sentences is the tokenization with Moses tokenizer [8] (Koehn et al., 2007).

For Chinese-English pair, we use the News Commentary v16. For preprocessing, we use Moses tokenizer to tokenize the English sentences and utilize the Jieba tool[9] (accurate mode) to tokenize the Chinese sentences in the parallel sentences and remove empty lines.

For English-Swedish, Finnish-Greek and Finnish-Hebrew pairs, we utilize subsets of CCMatrix v1.0 (Schwenk et al., 2021; Fan et al., 2021; Tiedemann, 2012). We use spacy to do the word segmentation for these language pairs.

The max number of unlabeled data that we use is shown in Table 11.

| Lang | Max Num |
|---|---|
| ja-en | 52,240 |
| de-en | 30,000 |
| ro-en | 30,000 |
| zh-en | 36,000 |
| en-sv | 30,000 |
| fi-el | 30,000 |
| fi-he | 30,000 |

Table 11: Size of parallel data.

## A.3 Training Details

For all supervised fine-tuning experiments, we used the PyTorch(2.4.0) (Paszke et al., 2019), transformers (4.45.2) (Wolf et al., 2020) and PEFT (0.12.0) libraries for loading model weights and training. We utilized vLLM[10] (version 0.6.2) (Kwon et al., 2023) to speed up inference and evaluation. Most experiments are conducted on a server with four NVIDIA RTX 6000 Ada GPUs. Our largest experiment takes approximately 60 hours of training time on a single NVIDIA RTX 6000 Ada GPU. For LLM-based word aligners, we mainly utilized LoRA (Hu et al., 2022) for parameter-efficient fine-tuning.

## A.4 LoRA training details

We use LoRA for decoder-based language models. The rank parameter is set to 64. The alpha parameter for LoRA scaling is set to 256. The dropout probability for LoRA layers is set to 0.05. The target modules for LoRA layers contains "gate_proj", "down_proj", "down_proj", "up_proj", "q_proj",

[7] http://web.eecs.umich.edu/~mihalcea/wpt/data.protected/Romanian-English.training.tar.gz

[8] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[9] https://github.com/fxsjy/jieba

[10] https://github.com/vllm-project/vllm

"v_proj", "k_proj" and "o_proj" for Llama and Aya series models.

### A.5 Evaluation Metric Details

Given a set of predicted alignment ($\mathcal{A}$), a set of sure alignment ($\mathcal{S}$) and a set of possible alignment ($P$), Precision, Recall, F1 and AER are computed as follows:

$$Precision(\mathcal{A}, \mathcal{P}) = \frac{|\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}|} \quad (2)$$

$$Recall(\mathcal{A}, \mathcal{S}) = \frac{|\mathcal{A} \cap \mathcal{S}|}{|\mathcal{S}|} \quad (3)$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (4)$$

$$AER(\mathcal{A}, \mathcal{S}, \mathcal{P}) = 1 - \frac{|\mathcal{A} \cap \mathcal{S}| + |\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}| + |\mathcal{S}|} \quad (5)$$

### A.6 Prompt Format and Example

Figure 6 is the prompt format of "marker mode". The model reponse contains a JSON output. The above panel shows the situation where the model outputs a JSON output when the marked source word has valid alignments. The below panel presents how the model deals with "no alignment" situation. Figure 4 shows a prompt example of "full mode". Figure 5 shows a prompt example of "marker mode".

## B Ablation Studies

**Multi-language Training.** We discuss whether combining the training data of all language pairs could further improve the word alignment performance of SemiAlign-D.

We mainly discuss two settings:

- Whether the combination of the labeled data from the four language pairs improves the performance of LLM-based word aligners or not.
- Whether combining labeled and unlabeled data from all language pairs based on the size of labeled data per each language pair improves the performance or not.

Table 12 shows the AER scores of single language pair (single-lang) and all language pair (multi-lang) training. We find that the results of using all training data are better if we only use the labeled data. However, if we add pseudo-labeled data to our training data, the results of separately training SemiAlign-D are better in most cases except the Chinese-English pair.

| SemiAlign-D | Ja-En | De-En | Ro-En | Zh-En |
|---|---|---|---|---|
| labeled only | | | | |
| single-lang | 16.16 | 11.93 | 14.66 | 5.73 |
| multi-lang | **14.47** | **9.21** | **10.61** | **4.54** |
| labeled + unlabeled | | | | |
| single-lang | **11.64** | **7.27** | **7.98** | 4.36 |
| multi-lang | 11.92 | 7.98 | 8.72 | **4.35** |

Table 12: AER results of single language training and multi-language training. Lower AER scores indicate better performance.

**Number of Iterations.** In this section, we discuss the effect of increasing iterations for self-training of SemiAlign-E. Figure 7 shows the AER scores of the first iteration and second iteration on the Japanese-English word alignment dataset. We observe that the second iteration of training provides limited improvement in word alignment performance, possibly due to the presence of noisy pseudo-labeled data.

**Effect of Using BinaryAlign-predicted Pseudo-labeled Data and Using SemiAlign-E-predicted labels.** Table 13 shows the results of SemiAlign-D on the Japanese-English dataset when using BinaryAlign-predicted labels and SemiAlign-E-predicted labels. We present the best results in their settings. We find that using SemiAlign-E-predicted labels is slightly better than using BinaryAlign-predicted labels.

| | Precision | Recall | F1 | AER |
|---|---|---|---|---|
| BinaryAlign | 88.53 | **87.99** | 88.26 | 11.74 |
| SemiAlign-E | **88.09** | 88.63 | **88.36** | **11.64** |

Table 13: Results of SemiAlign-D using different pseudo-labeled data. BinaryAlign means the results of using BinaryAlign-predicted labels. SemiAlign-E means the results of using SemiAlign-E-predicted labels

**Zero-shot Cross-lingual Transfer Between High-resource Languages.** In this section, we evaluate the zero-shot cross-lingual transfer performance of multilingual large language models as word aligners between high-resource languages. We use the SemiAlign-D model that is trained on one language pair, for example, Japanese-English or Chinese-English pair. Then we evaluate the model on the Romanian-English word alignment dataset directly.

**Prompt Example for LLM-based Word Alignment (Type: Full Mode)**

Role: system
Content: You are a helpful AI assistant for word alignment.

Role: user
Content: Please give me the word alignment information between the following two sentences. The parallel sentences:
足利_0 義満_1 （_2 あしかが_3 よしみつ_4 ）_5 は_6 室町_7 幕府_8 の_9 第_10 3_11 代_12 征夷_13 大_14 将軍_15 （_16 在位_17 1368_18 年_19 -_20 1394_21 年_22 ）_23 で_24 あ_25 る_26 。_27

yoshimitsu_0 ashikaga_1 was_2 the_3 3rd_4 seii_5 taishogun_6 of_7 the_8 muromachi_9 shogunate_10 and_11 reigned_12 from_13 1368_14 to1394_15 ._16

Role: assistant
Content: Here is the word alignment information:
```json
[{'足利_0': ['ashikaga_1']}, {'義満_1': ['yoshimitsu_0']}, {'あしかが_3': ['ashikaga_1']}, {'よしみつ_4': ['yoshimitsu_0']}, {'室町_7': ['muromachi_9']}, {'幕府_8': ['shogunate_10']}, {'の_9': ['of_7']}, {'第_10': ['the_3']}, {'3_11': ['3rd_4']}, {'代_12': ['3rd_4']}, {'征夷_13': ['seii_5']}, {'大_14': ['taishogun_6']}, {'将軍_15': ['taishogun_6']}, {'在位_17': ['reigned_12']}, {'1368_18': ['1368_14']}, {'年_19': ['1368_14']}, {'1394_21': ['to1394_15']}, {'年_22': ['to1394_15']}, {'で_24': ['was_2']}, {'あ_25': ['was_2']}, {'る_26': ['was_2']}, {'。_27': ['._16']}]
```.

Figure 4: Prompt example of the “full mode”.

Through this setting, we want to know the potential zero-shot cross-lingual transfer performance if our LLM-based word aligner SemiAlign-D is used on an unseen language pair.

Specifically, we trained the SemiAlign-D on the combination of the German-English labeled and pseudo-labeled data. Then we test its performance on Romanian-English word alignment test set.

The results are shown in Table 14. These results demonstrate that the pseudo-labeled data helps zero-shot cross-lingual transfer. We notice that the F1 and AER scores when we train models on De-En and test on Ro-En are larger than the ones of training models on Ja-En and testing on Ro-En. The reason might be that German is more similar to Romanian compared to Japanese in some linguistic aspects.

| SemiAlign-D | Ro-En (F1) | Ro-En (AER) |
|---|---|---|
| De-En (only labeled) | 82.45 | 17.55 |
| De-En (labeled + unlabeled) | **84.35** | **15.65** |
| Ja-En (only labeled) | 79.24 | 20.76 |
| Ja-En (labeled + unlabeled) | **80.20** | **19.80** |

Table 14: Result of zero-shot cross-lingual transfer on SemiAlign-D. The row name denotes the training language pair. The column name shows the test language pair. Higher F1 and lower AER scores indicate better performance.

**Effect of Possible Alignment.** Some word alignment datasets have possible alignments. For example, German-English word alignment dataset has possible alignments. In this section, we investigate the effect of training with possible alignment and without it using multilingual large language models.

Table 15 shows the results on the German-English dataset when using the labeled data with possible alignments and the labeled data without possible alignments. We find that the results without using possible alignment are better in F1 and AER scores.

| Precision | Recall | F1 | AER |
|---|---|---|---|
| w/ possible alignment | | | |
| 93.63 | **90.75** | 92.17 | 7.80 |
| w/o possible alignment | | | |
| **93.97** | 90.70 | **92.31** | **7.66** |

Table 15: Results of SemiAlign-D trained with the labeled data and pseudo-labeled data on German-English language pair. "w/ possible alignment" means the results of using possible alignments in the labeled data. "w/o possible alignment" means the results of not using possible alignment in the labeled data.

**Effect of Using LoRA, QLoRA and Full Parameter Fine-tuning.** Table 16 shows the results of using LoRA, QLoRA and Full parameter fine-tuning on the test set of Japanese-English word

Prompt Example for LLM-based Word Alignment (Type: Marker Mode)

Role: system
Content: You are a helpful AI assistant for word alignment.

Role: user
Content: Please give me the word alignment information between the following two sentences given the marked word with ¶. The parallel sentences:
¶ 足利_0 ¶ 義満_1 （_2 あしかが_3 よしみつ_4 ）_5 は_6 室町_7 幕府_8 の_9 第_10 3_11 代_12 征夷_13 大_14 将軍_15 （_16 在位_17 1368_18 年_19 -_20 1394_21 年_22 ）_23 で_24 あ_25 る_26 。_27

yoshimitsu_0 ashikaga_1 was_2 the_3 3rd_4 seii_5 taishogun_6 of_7 the_8 muromachi_9 shogunate_10 and_11 reigned_12 from_13 1368_14 to1394_15 ._16

Role: assistant
Content: Here is the word alignment information:

```json
{"足利_0": ["ashikaga_1"]}
```.

Figure 5: Prompt example of the "marker mode" for supervised fine-tuning of LLM-based word alignment models. The input is the marked source sentence and target sentence.

alignment dataset, KFTT. We find that the LoRA verison achieves the best performance.

**Prompt Format for LLM-based Word Alignment (Type: Marker Mode)**

Role: system
Content: You are a helpful AI assistant for word alignment.

Role: user
Content: Please give me the word alignment information between the following two sentences given the marked word with {context_sep}. The parallel sentences: {marked_sent_src}\n\n\n{sent_tgt_with_id}

Role: assistant
Content: Here is the word alignment information:\n\n```json\n{label_str}\n```.

**Prompt Format for LLM-based Word Alignment (Type: Marker Mode)**

Role: system
Content: You are a helpful AI assistant for word alignment.

Role: user
Content: Please give me the word alignment information between the following two sentences given the marked word with {context_sep}. The parallel sentences: {marked_sent_src}\n\n\n{sent_tgt_with_id}

Role: assistant
Content: There is no word alignment information for the marked word.

Figure 6: Prompt format of "marker mode" for supervised fine-tuning of LLM-based word alignment models. The input is the marked source sentence and target sentence.

| Method | Precision | Recall | F1 | AER |
|---|---|---|---|---|
| SemiAlign-D (LoRA) | 88.09 | 88.63 | 88.36 | 11.64 |
| SemiAlign-D (QLoRA) | 87.82 | 88.48 | 88.15 | 11.85 |
| SemiAlign-D (Full) | 89.12 | 87.18 | 88.14 | 11.86 |

Table 16: Results of using LoRA, QLoRA and "full parameter fine-tuning" on the test set of Japanese-English word alignment dataset, KFTT.

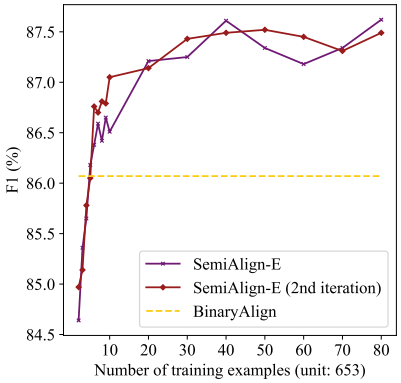


Figure 7: Comparison between the F1 scores of the SemiAlign-E after the first iteration and second iteration training on the Japanese-English dataset. Higher F1 scores mean better performance.