BinaryAlign: Word Alignment as Binary Sequence Labeling

Gaetan Lopez Latouche and **Marc-André Carbonneau** and **Ben Swanson**
Ubisoft La Forge
{gaetan.lopez-latouche,marc-andre.carbonneau2,ben.swanson2}@ubisoft.com

Abstract

Real world deployments of word alignment are almost certain to cover both high and low resource languages. However, the state-of-the-art for this task recommends a different model class depending on the availability of gold alignment training data for a particular language pair. We propose BinaryAlign, a novel word alignment technique based on binary sequence labeling that outperforms existing approaches in both scenarios, offering a unifying approach to the task. Additionally, we vary the specific choice of multilingual foundation model, perform stratified error analysis over alignment error type, and explore the performance of BinaryAlign on non-English language pairs. We make our source code publicly available.[1]

1 Introduction

Word alignment refers to the task of uncovering word correspondences between translated text pairs. The automatic prediction of word alignments dates back to the earliest work in machine translation with the IBM models (Brown et al., 1993) where they were used as hidden variables that permit the use of direct token to token translation probabilities. While state of the art machine translation techniques have largely abandoned the use of word alignment as an explicit task (Li, 2022) other use cases for alignments have emerged including lexical constraint incorporation (Chen et al., 2021b), analysing and evaluating translation models (Bau et al., 2018; Neubig et al., 2019), and cross-lingual language pre-training (Chi et al., 2021b).

In many real-world applications word alignment must be performed across several languages, often including languages with manually annotated word alignment data and others lacking such annotations. We refer to those languages as high and low-resource languages respectively. While word alignment for high-resource languages can be learned in a few-shot or fully supervised setting depending on the amount of data, for low-resource languages zero-shot learning strategies must be employed due to data scarcity.

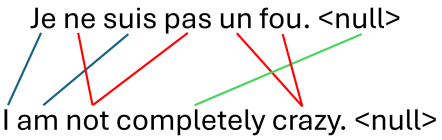

Figure 1: Example of alignment of an approximate translation, as often encountered in real-world applications. Links in red indicate situations where one word is aligned with several contiguous or non-contiguous words. The green line represent a situation where a word is untranslated which happens in many language pairs.

State-of-the-art supervised techniques formalize the task of word alignment as a collection of SQuAD-style span prediction problems (Nagata et al., 2020; Wu et al., 2023) while in zero-shot settings the best performing methods induce word alignment from the contextualized word embeddings of mulitingual pre-trained language models (mPLMs) (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Wang et al., 2022). From a practical perspective, this discrepancy in the preferred method adds complexity to the deployment of word alignment models in real-world applications where both high and low-resource languages must be supported.

We observe a deeper issue that both span prediction and contextualized word embeddings are sub-optimal as each induces a bias in word alignment models that limits their accuracy. Span prediction methods cannot robustly deal with discontinuous word alignments without relying on complex post-processing and hyper-parameter tuning. Contextualized word embeddings method cannot deal effectively with untranslated words and one-to-

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

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10277–10288

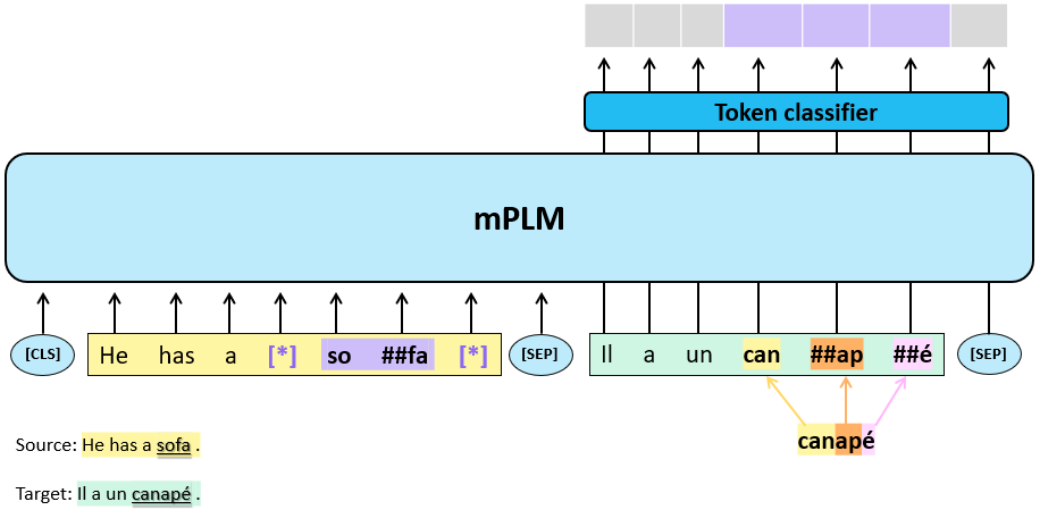

Figure 2: Illustration of our formalization of word alignment. In this example, the word "sofa" in the source sentence and the word "canapé" in the target sentence are aligned.

multiple alignments because they rely on a softmax function that normalizes predictions at a sentence-level while in word alignment; one token being aligned to token T does not mean that another token is less likely to be aligned to T. This poses word alignment as a single-label classification problem, while in reality it is better viewed as a series of binary classifications applied to each possible pair of words. Figure 1 shows some cases of one-to-multiple alignments, non-contiguous spans and untranslated words.

In this paper, we present BinaryAlign, a novel word alignment solution that outperforms the state-of-the-art in zero-shot, few-shot and fully-supervised settings. In particular, we reformulate word alignment as a set of binary classification tasks in which an individual alignment prediction is made for each possible pair of words. This reformulation of the task outperforms all previous approaches over five different language pairs with varying levels of supervision.

2 Related Work

Recently, mPLM based approaches have significantly outperformed bilingual statistical methods (Och and Ney, 2003; Dyer et al., 2013; Östling and Tiedemann, 2016) and bilingual neural methods (Garg et al., 2019; Zenkel et al., 2020; Chen et al., 2020, 2021a; Zhang and van Genabith, 2021). Among those approaches, we distinguish methods that achieve good performance without relying on manually annotated word alignment datasets (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Wang et al., 2022) from supervised methods that leverage existing word alignment datasets to train high performing word aligners (Nagata et al., 2020; Wu et al., 2023).

The first type of method relies on the approach of SimAlign (Jalili Sabet et al., 2020) which proposes to induce word alignment from the contextualized word embeddings of mPLMs pre-trained on non-parallel data. AwesomeAlign (Dou and Neubig, 2021) builds on top of this approach and proposes to fine-tune mPLMs on parallel text with different objectives to improve the quality of the contextualized word embeddings. More recently, AccAlign (Wang et al., 2022) showed that models trained to learn language-agnostic sentence-level embeddings also learn strong language-agnostic word-level embeddings and set the state of the art in the zero-shot setting. Also, Wang et al. (2022) show that fine-tuning on existing word alignment datasets improves performance of AccAlign on language pairs unseen during word alignment fine-tuning. Our method is different from this body of work because our training and inference objective differ. We formalize word alignment as a binary sequence labeling task while we can see those methods as framing word alignment as a token retrieval task.

In terms of approaches trained and evaluated in a supervised setting SpanAlign (Nagata et al., 2020) formalizes word alignment as a collection of

SQuAD-style span prediction problems which differs from our binary classification objective. However, this method falls short in zero-shot and few-shot when word alignment training data is scarce. To remedy this problem, WSPAlign (Wu et al., 2023) introduces a pre-training method based on weak supervision that significantly improves performance for all amounts of training data.

3 Method

Given a sentence $\mathcal{X}$ with n words and a translation into another language $\mathcal{Y}$ with m words, the task of word alignment is to produce an n by m adjacency matrix for the bipartite graph with the words of $\mathcal{X}$ on one side and the words of $\mathcal{Y}$ on the other (refer to Figure 1 for an illustration). As our model will employ commonly used subword tokenization preprocessing we assume access to an invertible tokenizer, often implemented in practice with a list of subword units, a greedy algorithm for subword chunking, and leading symbols to denote word continuation in the vocab file (Sennrich et al., 2016; Wu et al., 2016).

We present BinaryAlign, a novel word alignment approach using a binary sequence labeling model, shown in Figure 2. The inputs to this model are a subword tokenized source sentence $X = x_1, x_2, ..., x_{|X|}$, a subword tokenized target sentence $Y = y_1, y_2, ..., y_{|Y|}$, and a reference word $w_X = x[i:j]$ which is a subspan of X. We model the distribution of a binary alignment vector A of size $|Y|$ in which each entry a_k indicates if the word in Y that contains y_k is aligned to w_X.

We first preprocess X by surrounding w_X with unique separator tokens and then cross-encode the source and target sentences with an mPLM. For each token y_k in the target sentence we pass its final encoded representation through a linear layer to produce a single logit z_k. We model a_k with a logistic function using z_k as its parameter:

$$p(a_k = 1|w_X) = \frac{1}{1 + e^{-z_k}} \quad (1)$$

A supervised signal for token level alignments a_k is easily divined from word alignment data and so this form is sufficient to estimate the parameters of the model. However, our true inference-time goal of word to word alignment requires the use of additional heuristics. To motivate these heuristics we formalize $\mathcal{W}$ as the inverse of the surjective mapping between the token indices in Y and its corresponding words; for any subspan w_Y of Y, $\mathcal{W}(w_Y)$ returns the token indices in Y that compose w_Y.

Given an aggregation function agg, we define the probability of the event a' that there exists an alignment between word w_X in X and word w_Y in Y as

$$p(a') = \underset{\forall k \in \mathcal{W}(w_Y)}{agg} p(a_k = 1|w_X) \quad (2)$$

Preliminary experiments suggest that the maximum aggregation strategy yields slightly superior performance compared to the mean and minimum aggregation strategies; hence, we adopt it for all subsequent experiments.

Word alignment in its general form is a symmetric problem in that we would expect the same answer if the source and target were swapped. However, like most leading word alignment methods, our method is asymmetric; the source and target sentences are handled differently. This deficiency is empirically detrimental to performance with the common remedy being to perform alignment in both directions and then to merge the two predictions in some manner.

We use the following symmetrization technique: letting $p_{X \to Y}$ denote the use of X as the source and Y as the target sentence, we average $p_{X \to Y}(a')$ and $p_{Y \to X}(a')$ and apply a threshold decision rule to make our final inference prediction. While outside the scope of this study, we note that various other options exist and have been explored in previous work such as bidirectional average (Nagata et al., 2020) or intersection, union and grow-diag-final: the default symmetrization heuristics supported in Moses (Koehn et al., 2007).

4 Experiments

4.1 Datasets

We use seven datasets of manually annotated word alignment data for our main experiments: French-English (fr-en), Chinese-English (zh-en), Romanian-English (ro-en), Japanese-English (ja-en), German-English (de-en), Swedish-English (sv-en) and ALIGN6. The ja-en data comes from the KFTT word alignment data (Neubig, 2011), while the ro-en and fr-en data are taken from Mihalcea and Pedersen (2003) and the de-en data is provided by Vilar et al. (2006). Also, the zh-en data is

	de-en	ro-en	fr-en	zh-en	ja-en	avg
Bilingual Methods			AER(%)			
FAST-ALIGN (DYER ET AL., 2013)	27.0	32.1	10.5	38.1	51.1	31.8
GIZA++ (OCH AND NEY, 2003)	20.6	26.4	5.9	35.1	48.0	27.2
EFLOMAL (ÖSTLING AND TIEDEMANN, 2016)	22.6	25.1	8.2	28.7	47.5	26.4
MASK-ALIGN (CHEN ET AL., 2021A)	14.4	19.5	4.4	-	-	-
BTBA (ZHANG AND VAN GENABITH, 2021)	14.3	18.5	6.7	-	-	-
Multilingual Methods			AER(%)			
SIMALIGN (JALILI SABET ET AL., 2020)	18.8	27.2	7.6	21.6	46.6	24.4
WSP (WU ET AL., 2023)	16.4	20.7	9.0	21.6	43.0	22.1
AWESOMEALIGN (DOU AND NEUBIG, 2021)	15.2	25.5	4.0	13.4	40.6	19.7
SPANALIGN-ALIGN6	13.3	25.9	2.9	15.5	41.0	19.7
ACCALIGN (WANG ET AL., 2022)	13.5	20.8	2.8	11.3	37.0	17.1
BINARYALIGN	**11.6**	**19.1**	**1.5**	**9.0**	**29.2**	**14.1(↓3.0)**

Table 1: Comparison of AER(%) between our method (BinaryAlign) and previous works on five unseen word alignment language pairs (zero-shot cross-lingual transfer). We highlight the best performance for each language pair in bold font. The arrow shows the performance improvement when compared to previous state-of-the-art.

obtained from the TsinghuaAligner website[2] and the sv-en dataset[3] from Holmqvist and Ahrenberg (2011). Finally, ALIGN6 (Wang et al., 2022) is the combination of six different word alignment datasets featuring Dutch-English (Macken, 2010), Czech-English (Mareček, 2011), Hindi-English (Aswani and Gaizauskas, 2005), Turkish-English (Cakmak et al., 2012), Spanish-English (Graca et al., 2008) and Portuguese-English (Graca et al., 2008).

In addition, we use the Finnish-Greek (fi-el) and the Finnish-Hebrew (fi-he) word alignment test dataset from Imani et al. (2021) to experiment on non-English language pairs. Note that all datasets are the same ones used in Wang et al. (2022).

4.2 Experimental setup

Unseen alignment experiments: In our unseen alignment experiments, models are not fine-tuned on manual word alignment data of the tested language pair. This replicates a common real-world situation in which alignment data set is not available for a language pair and models must leverage knowledge gleaned from other language pairs and pre-training. This setting is usually referred as zero-shot cross-lingual transfer (Conneau et al., 2020; Chi et al., 2021a). We follow Wang et al. (2022) and fine-tune our model on ALIGN6 and use sv-en as our validation set. We evaluate our method following the experimental protocol of previous work (Dou and Neubig, 2021; Wang et al., 2022) and use de-en, ro-en, fr-en, zh-en and ja-en for testing. Note that those language pairs are not included in ALIGN6.

Few-shot and fully supervised experiments: We follow the protocol of Wu et al. (2023) for our few-shot and fully supervised experiments on de-en, ro-en, fr-en and ja-en. For ja-en, we train on all eight dev set files, we use four test set files for testing, and the remaining three test files for validation. We separate the de-en, ro-en and fr-en data into training and test sets. We fine-tune using 300 sentences for fr-en and de-en, while we use 150 sentences for ro-en. All remaining sentences are used for testing. Note that we made the same splits[4] as Wu et al. (2023). For the zh-en data, we leverage the datasets provided in v1 of the TsinghuaAligner website. We use their dev set and test set as our training and test set respectively. Both contain 450 sentences.

In our non-English experiments, we evaluate on Finnish to Greek (fi-el) and Finnish to Hebrew (fi-he) data. For fi-el we use 400 samples for training and test on the 391 remaining samples, and for fi-he, we have 1780 samples for training and 450 for test. We use 32 training samples for all our few-shot experiments.

A detailed account of the number of sentence pairs for each dataset and settings used in our experiments is available in appendix.

4.3 Baseline methods

Unseen alignment experiments: In this setting, we compare BinaryAlign to the three main bodies

[2] http://nlp.csai.tsinghua.edu.cn/~ly/systems/TsinghuaAligner/TsinghuaAligner.html

[3] https://www.ida.liu.se/divisions/hcs/nlplab/resources/ges/

[4] https://huggingface.co/datasets/qiyuw/wspalign_ft_data

	de-en	ro-en	fr-en	zh-en	ja-en	avg
Few-Shot Supervision			AER(%)			
AccAlign (Wang et al., 2022)	11.9	16.5	2.7	10.7	35.3	15.4
SpanAlign (Nagata et al., 2020)	15.4	14.8	8.0	16.0	43.3	19.5
WSPAlign (Wu et al., 2023)	10.2	10.9	3.8	11.1	28.2	12.8
BinaryAlign-noPre	9.6	10.1	5.1	8.6	25.3	11.7
BinaryAlign	**7.6**	**8.8**	**2.5**	**6.7**	**22.8**	**9.7(↓3.1)**
Full Supervision			AER(%)			
AccAlign (Wang et al., 2022)	11.7	16.8	2.6	10.1	31.2	14.5
SpanAlign (Nagata et al., 2020)	14.4	12.2	4.0	8.9	22.4	12.4
WSPAlign (Wu et al., 2023)	11.1	8.6	2.5	7.6	16.3	9.2
BinaryAlign-noPre	8.0	7.8	**1.7**	5.2	14.2	7.4
BinaryAlign	**7.7**	**7.3**	1.9	**4.8**	**13.9**	**7.1(↓2.1)**

Table 2: Comparison of AER(%) between the proposed method (BinaryAlign) and previous works with few-shot and full supervision. We highlight in bold the best performance in each problem. The arrow shows the performance improvement over previous state-of-the-art.

of research that evaluate on unseen word alignment language pairs. The first one corresponds to the historical bilingual statistical methods. We report GIZA++ (Och and Ney, 2003), eflomal (Östling and Tiedemann, 2016) and fast-align (Dyer et al., 2013) which are the best-known statistical methods.

Bilingual neural methods represent the second body of work. For this, we report MASK-ALIGN (Chen et al., 2021a) and BTBA-FCBO-SST (Zhang and van Genabith, 2021), the two best performing bilingual neural methods.

Finally, we compare to three methods relying on contextualized word embeddings of mPLMs: SimAlign (Jalili Sabet et al., 2020), AwesomeAlign (Dou and Neubig, 2021) and AccAlign (Wang et al., 2022). Note that AccAlign leverages ALIGN6 and is the state-of-the-art method for unseen alignment.

We reimplemented AccAlign using their adapter method on ALIGN6 as done in there paper. For other baselines, we quote the results from Dou and Neubig (2021) for bilingual statistical methods, from Wang et al. (2022) for AwesomeAlign, SimAlign, and the bilingual neural methods. We also report the zero-shot performance of WSPAlign (Wu et al., 2023) that we computed using the checkpoints provided by the authors. Also, we train SpanAlign on ALIGN6, the same dataset that is used to train AccAlign and BinaryAlign in this setup. Comparing to this baseline allows us to determine if our formalization of word alignment can better leverage existing word alignment datasets than SQuAD-style span prediction techniques.

Few-shot and fully supervised experiments: We compare to SpanAlign (Nagata et al., 2020) and WSPAlign (Wu et al., 2023), the two state of the art supervised word alignment techniques. Also, we further fine-tune AccAlign on our language specific training sets. Comparing to this baseline is important as it allows us to determine if our proposed method performs better than state of the art contextualized word embedding extraction techniques when we have access to manual word alignments of the same language pair. Also, SpanAlign and AccAlign have not been compared in previous studies.

We reimplemented WSPAlign and SpanAlign for all our few-shot experiments using the source code provided by WSPAlign authors[5]. We quote the results from Wu et al. (2023) for supervised fine-tuning of SpanAlign and WSPAlign on de-en, ro-en, ja-en, fr-en and reimplement them for zh-en as we use a different train and test dataset than the original papers.

4.4 Fine-tuning setups

We did not perform extensive hyper-parameter tuning of our methods. We arbitrarily use a learning rate of $2e^{-5}$, a batch size of 8 for fine-tuning and pre-training, and a threshold of 0.5 for inference. We train all our models for 5 epochs, except for few-shot learning without pre-training on ALIGN6 where we train for 25 epochs. Results would improve with hyper-parameter tuning on a large validation set. We use mdeberta-v3-base (He et al., 2021) as our mPLM. We discuss the choice of mPLM in section 4.6.2. In supervised set-

[5]https://github.com/qiyuw/WSPAlign

		fi-el	fi-he
Unseen alignment	AccAlign	37.0	73.7
	WSPAlign	34.9	74.7
	BinaryAlign	**24.3**	**40.4**
few-shot	AccAlign	30.3	51.5
	WSPAlign	14.8	27.0
	BinaryAlign	**10.8**	**16.9**
full	AccAlign	25.1	40.15
	WSPAlign	8.6	10.8
	BinaryAlign	**6.2**	**7.2**

Table 3: Comparison of AER(%) between our method (BinaryAlign) and previous works in different settings on two non-English language pairs.

tings we report both BinaryAlign and BinaryAlign-noPre, our method with and without pre-training on ALIGN6. Note that pre-training our model on ALIGN6 is the default setting in BinaryAlign.

4.5 Word Alignment Evaluation Metric

Following previous works (Wang et al., 2022; Dou and Neubig, 2021), we evaluate performance using Alignment Error Rate (AER). Given a set of sure alignments (S), possible alignments (P) and predicted alignments (H) we can calculate AER as follows:

$$AER(S, P, H) = 1 - \frac{|H \cap S| + |H \cap P|}{|H| + |S|}.$$

Following the protocol in Wu et al. (2023) we use only sure alignments for training but we evaluate on both sure and possible alignments when the distinction is available.

4.6 Results and Discussion

We performed several experiments to validate BinaryAlign. First we compare our method to other state-of-the-art methods in three different levels of supervision: full available supervision, few-shot (32 samples), and unseen languages (zero-shot cross-lingual transfer). We also evaluate on non-English language pairs. Next, we evaluate the impact of choices for mPLM foundation and symmetrization. Finally, we discuss how our problem formulation compares to span prediction and contextualized word embedding based approaches in different situations.

4.6.1 Comparison to State-of-the-Art

Unseen alignments: As a first experiment, we apply all methods to new language pairs without performing word alignment fine-tuning on the tested language pair as explained in 4.2. Table 1 reports the AER of all methods. BinaryAlign is the new state-of-the-art on all language pairs. In particular, it outperforms AccAlign by 3.0 points of AER on average. Since AccAlign and BinaryAlign share the same pre-training dataset (ALIGN6), this indicates that our word alignment problem formulation performs better than inducing word alignment from contextualized word embeddings. This is also true when comparing with SpanAlign pre-trained on the same data (ALIGN6). This indicates that our formalization of word alignment promotes learning more language-agnostic signals from word alignment datasets when compared to existing methods.

Full and few shot supervision: We compare BinaryAlign to the other baseline methods after fine-tuning on alignment data with few samples and the whole training data set. Table 2 shows the results for both supervision levels.

Our method achieves new state-of-the-art on all tested language pairs and with both levels of supervision. On average it outperforms WSPAlign, the previous state-of-the-art, by 2.1 points of AER with full supervision and 3.1 with few-shot supervision.

Even without pre-training on ALIGN6 our method outperforms all methods. Given that WSPAlign was pre-trained on 2 millions samples, it indicates that BinaryAlign promotes sample efficiency.

Finally, we highlight that by using only 32 samples for few-shot supervision BinaryAlign outperforms SpanAlign regardless of pre-training. This reinforces the performance improvement of formalizing word alignment as a binary token classification objective over span prediction.

Impact of pre-training on other languages: Table 2 reports the results of our method with and without pre-training on ALIGN6. We conclude that pre-training improves AER with few-shot as well as full supervision. However, we observe a smaller improvement with full supervision which suggests that the benefit of pre-training on other languages is inversely correlated with the amount of in-domain word alignment data. While pre-training encourages sample efficiency, we did not find any indication that it could hinder performance.

Non-English language pairs: Because usually mPLMs perform better in English it is important to investigate how our method performs on non-English language pairs. Table 3 reports results on

	de-en	ro-en	fr-en	zh-en	ja-en	avg
Few-shot Supervision			AER(%)			
mBERT	9.8	12.7	3.5	9.3	26.6	12.4
mDeBERTa	**7.6**	8.8	2.5	6.7	22.8	9.7
LaBSE	7.9	9.3	2.4	6.4	23.4	9.9
XLM-RoBERTa-base	8.4	9.3	**2.4**	8.5	31.1	11.9
XLM-RoBERTa-large	7.7	**8.4**	3.1	**6.0**	**21.8**	**9.4**
Full Supervision			AER(%)			
mBERT	9.4	10.3	3.1	6.6	16.8	9.2
mDeBERTa	**7.7**	7.3	**1.9**	4.8	13.9	7.1
LaBSE	**7.7**	7.3	2.4	5.1	14.7	7.4
XLM-RoBERTa-base	8.4	7.4	2.3	5.9	16.7	8.1
XLM-RoBERTa-large	**7.7**	**6.9**	2.2	**4.4**	**13.6**	**7.0**

Table 4: Comparison of AER(%) of BinaryAlign under few-shot and full supervision using different mPLMs.

Test set	Direction	SpanAlign	BinaryAlign
de-en	de-en	83.6(↓2.0)	91.9(↓0.4)
	en-de	84.5(↓1.1)	92.2(↓0.1)
	sym	85.6	92.3
ro-en	ro-en	85.5(↓2.3)	92.6(↓0.1)
	en-de	86.7(↓1.1)	92.2(↓0.5)
	sym	87.8	92.7
fr-en	fr-en	85.0(↓1.7)	98.0(↓0.2)
	en-fr	85.4(↓1.3)	98.0(↓0.2)
	sym	86.7	98.2
ja-en	ja-en	80.2(↑2.4)	85.6(↓0.5)
	ja-en	65.2(↓12.4)	85.6(↓0.5)
	sym	77.6	86.1

Table 5: Comparison of our method (BinaryAlign) and reported results for SpanAlign (Nagata et al., 2020) when using symmetrization. We report the F1 score for each direction and the best symmetrized result (sym) from all explored heuristics. See appendix for details on our results and metric.

language pairs excluding English. We used the checkpoint[6] provided by the authors of WSPAlign since paragraph pairs in Finnish-Greek and Hebrew are difficult to obtain for training. Our results show that BinaryAlign outperforms WSPAlign and AccAlign for all degree of supervision. Also, the AER in non-English language pairs seems to be similar to the AER of our main experiments on English-centric language pairs which shows that our method does not depend on English and is robust to variations in language family.

4.6.2 Design Choices

mPLM Architecture: Our proposed reformulation of the word alignment problem does not depend on a particular mPLM architecture. In this experiment, we investigate the impact of using different mPLMs. We explore five different mPLMs: XLM-RoBERTa (base and large)(Conneau et al., 2020), LaBSE[7] (Feng et al., 2022), mDeBERTa-v3-base[8] (He et al., 2021) and mBERT[9] (Devlin et al., 2019).

Table 4 reports AER of BinaryAlign using different mPLMs in few-shot and fully supervised settings. All these versions of BinaryAlign reach or surpass the previous state-of-the-art in terms of average AER on the five tested language pairs. This highlights that the improvement of our method over previous state-of-the-art is not explained by its reliance on a specific mPLM.

While most mPLMs yield similar results, mBERT performs slightly worse than the others. This could be due to a poor parametrization given that we used the same hyper-parameter configuration for all mPLMs. This could also be explained by the training objective of the mPLMs or their capacity. For example we observe that scaling the size of XLM-RoBERTa has an effect on alignment performance. The base model has an approximately similar capacity as the other mPLMs and when we increase this capacity using the large model we obtain the best result over all mPLMs. We suspect that this effect could generalize to other mPLM architectures.

Symmetrization: Here we investigate the impact of different symmetrization heuristics on our results. As stated in Section 3 symmetrization consists in fusing the alignment obtained going from one language to another with the alignment obtained going in the inverse direction. We com-

[6]https://huggingface.co/qiyuw/WSPAlign-xlm-base

[7]https://huggingface.co/sentence-transformers/LaBSE

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

[9]https://huggingface.co/bert-base-multilingual-cased

	de-en	ro-en	fr-en	zh-en	ja-en	avg
Untranslated words	Correctly aligned words(%)					
Number of occurances	2085	974	674	5882	6204	3164
AccAlign	74.1	79.5	82.2	75.9	75.0	77.3
SpanAlign	81.5	85.2	88.6	79.3	86.6	84.3
BinaryAlign	**84.4**	**88.3**	**94.1**	**83.5**	**89.0**	**87.9**
One-to-multiple alignments	Correctly aligned words(%)					
Number of occurances	2079	1726	6159	1738	3937	3128
AccAlign	13.8	5.3	1.2	44.9	11.8	15.4
SpanAlign	27.6	11.0	**5.6**	48.8	21.6	22.9
BinaryAlign	**31.4**	**16.7**	4.8	**60.2**	**29.0**	**28.4**
One-to-multiple non-contiguous words	Correctly aligned words(%)					
Number of occurances	383	410	565	179	405	388
AccAlign	5.5	5.6	3.5	15.1	4.9	6.9
SpanAlign	11.0	2.2	2.5	8.9	4.7	5.9
BinaryAlign	**21.7**	**5.1**	**7.1**	**26.3**	**7.4**	**13.5**

Table 6: Comparison of AccAlign, SpanAlign and BinaryAlign in complex word alignment situations. The three methods are pre-trained on ALIGN6 and evaluated on unseen alignments. See A.3 for details on our metric.

pare several symmetrization techniques: intersection, union, average (avg) and bidirectional average (bidi-avg) (Nagata et al., 2020).

In Table 5 we compare results obtained from aligning in a single direction to the results obtained using the best symmetrization heuristics (full details available in the appendix). We report the F1 score (see A.3) as done in Nagata et al. (2020).

Our results indicate that for BinaryAlign, unidirectional alignment does not perform significantly worse (average of 0.3 points of F1 score) than symmetrized alignment. This is not the case for SpanAlign which gains 2.4 points of F1 score on average by applying symmetrization. Performing alignment in only one direction is interesting since it halves the inference time.

4.6.3 Post-analysis of errors

In this section we analyze how the proposed problem formulation of BinaryAlign improves accuracy in complex word alignment situations. We inspect results in three situations: 1) words that are untranslated, also referred as null words (Jalili Sabet et al., 2020) (2) words that are aligned to multiple words (3) words that are aligned to multiple non contiguous words. For each situation, we report the percentage of correctly aligned words in Table 6. Details on how we computed our metric can be found in A.3. Results indicate that our method handles these situations better than both competing methods. This is especially true when aligning multiple non contiguous words which was the main motivation for our reformulation. The prevalence of these situations in a given language pair modulates the performance gain of our method over the others.

5 Conclusion

We presented BinaryAlign, a novel word alignment training and inference procedure. In particular, we proposed to reformulate the word alignment problem as a binary token classification task. We showed that because of this reformulation BinaryAlign outperforms existing methods regardless of the degree of supervision. In addition we showed that it overcomes the inherent limitations of previous methods relying on span prediction and softmax. As a result, we made the word alignment task easier to tackle by using a single model for both high and low-resource languages.

In the future we plan to explore the use of larger decoder-only or encoder-decoder models such as mT5 (Xue et al., 2021) to see how much alignment performance will increase. We also plan on investigating knowledge distillation techniques to improve the inference time of our method.

6 Limitations

The inference cost is the main limitation of our method. When using symmetrization, it has to perform a forward pass for each word of both sentences, which can be slow with long sequences.

However, this is a drawback that we share with previous state-of-the-art supervised approaches (Nagata et al., 2020; Wu et al., 2023).

In addition, we did not experiment on extremely low-resource languages that the mPLM has not seen during pre-training (Ebrahimi et al., 2023). While the benefits of our new formulation would likely apply to any language, it is unclear how our method will rapidly adapt the mPLM to new languages (Garcia et al., 2021).

In real-world applications, translations are often partial and noisy. Unfortunately, we could not evaluate the robustness of our method to different translation pair quality because this type of word alignment dataset does not exist.

7 Ethics Statement

The ethical and societal implications of word alignment and neural machine translation are generally positive. They facilitate cross-cultural communication and break down language barriers. Nonetheless, as for any machine learning field, potential biases embedded in training data can inadvertently influence translations and perpetuate stereotypes, especially when translating to and from gendered languages. Finally the impact of neural machine translation on employment for human translators raises questions about job displacement and economic inequalities. It is crucial for developers and stakeholders to prioritize fairness, transparency, and accountability in the design and implementation of such systems. We must balance technological advancement and ethical responsibility to ensure societal well-being, inclusive communication and minimize unintended consequences.

References

Niraj Aswani and Robert Gaizauskas. 2005. Aligning words in english-hindi parallel corpora. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 115–118.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Robert L Mercer, et al. 1993. The mathematics of statistical machine translation: Parameter estimation.

Mehmet Talha Cakmak, Süleyman Acar, and Gülsen Eryigit. 2012. Word alignment for english-turkish language pair. In *LREC*, pages 2177–2180.

Chi Chen, Maosong Sun, and Yang Liu. 2021a. Mask-align: Self-supervised neural 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 4781–4791, Online. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, and Victor OK Li. 2021b. Lexically constrained neural machine translation with explicit alignment guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12630–12638.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. 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.

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.

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.

Abteen Ebrahimi, Arya D. McCarthy, Arturo Oncevay, John E. Ortega, Luis Chiruzzo, Gustavo Giménez-Lugo, Rolando Coto-Solano, and Katharina Kann. 2023. Meeting the needs of low-resource languages: The value of automatic alignments via pretrained models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3912–3926, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Xavier Garcia, Noah Constant, Ankur P Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. *arXiv preprint arXiv:2103.06799*.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Joao Graca, Joana Paulo Pardal, Luísa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *LREC*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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.

Ayyoob Imani, Masoud Jalili Sabet, Lütfi Kerem Şenel, Philipp Dufter, François Yvon, and Hinrich Schütze. 2021. Graph algorithms for multiparallel word alignment. *arXiv preprint arXiv:2109.06283*.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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. Association for Computational Linguistics.

Bryan Li. 2022. Word alignment in the era of deep learning: A tutorial. *arXiv preprint arXiv:2212.00138*.

Lieve Macken. 2010. An annotation scheme and gold standard for dutch-english word alignment. In *7th conference on International Language Resources and Evaluation (LREC 2010)*, pages 3369–3374. European Language Resources Association (ELRA).

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

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.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *arXiv preprint arXiv:1903.07926*.

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

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

David Vilar, Maja Popović, and Hermann Ney. 2006. Aer: Do we need to "improve" our alignments? In *Proceedings of the Third International Workshop on Spoken Language Translation: Papers*.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Jingyi Zhang and Josef van Genabith. 2021. A bidirectional transformer based alignment model for unsupervised 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 283–292.

A Appendix

A.1 Experimental Environment

For all our experiments we use one NVIDIA Quadro RTX 6000. Fine-tuning on ALIGN6 took 4 hours and 30 minutes while our fully supervised experiments took on average 20 minutes per dataset.

A.2 Dataset statistics

Table 8 shows the number of samples that our training, validation and test set contains for all level of supervision. All dataset are the same as in Wang et al. (2022). The de-en, ro-en, fr-en and ja-en train-test splits are the same as the one used in Wu et al. (2023). We could not get the zh-en data used in Wu et al. (2023) because the dataset is not publicly available.

A.3 Metric details

A.3.1 F1 score

Given a set of sure alignments (S), possible alignments (P) and predicted alignments (H), we can compute the Recall, Precision and F1 score as follows:

$$Recall(H,S) = \frac{|H \cap S|}{|S|}$$

$$Precision(H,P) = \frac{|H \cap P|}{|H|}$$

$$F_1(H,S,P) = \frac{2 * Precision * Recall}{Precision + Recall}$$

When $S == P$, we have:

$$AER(H,S,P) = 1 - F_1(H,S,P)$$

A.3.2 Post-analysis of errors

Untranslated words: We report the number of untranslated words correctly aligned by the models over the total number of untranslated words. We consider a word to be correctly aligned if the model has not aligned it to any words in the corresponding translated sentence.

One-to-multiple contiguous and non contiguous words: In this case, we report the number of contiguous/non contiguous words correctly aligned by the model over the total number of contiguous/non contiguous words. We consider a word to be correctly aligned if the model has aligned it to the exact same set of ground truth aligned words.

Test set	Method	SpanAlign	BinaryAlign
de-en	De to En	83.6(↓2.0)	91.9(↓0.4)
	En to DE	84.5(↓1.1)	92.2(↓0.1)
	intersection	84.0	92.2
	union	84.0	91.9
	bidi-avg	85.6	92.2
	avg	-	92.3
ro-en	Ro to En	85.5(↓2.3)	92.6(↓0.1)
	En to Ro	86.7(↓1.1)	92.2(↓0.5)
	intersection	87.3	92.2
	union	85.0	92.7
	bidi-avg	87.8	92.7
	avg	-	92.7
fr-en	Fr to En	85.0(↓1.7)	98.0(↓0.2)
	En to Fr	85.4(↓1.3)	98.0(↓0.2)
	intersection	86.7	97.8
	union	83.9	98.2
	bidi-avg	86.2	97.8
	avg	-	98.0
ja-en	Ja to En	80.2(↑2.4)	85.6(↓0.5)
	En to Ja	65.2(↓12.4)	85.6(↓0.5)
	intersection	74.5	85.7
	union	71.1	85.5
	bidi-avg	77.6	85.7
	avg	-	86.1

Table 7: F1 score comparison of our method (BinaryAlign) and SpanAlign using different symmetrization heuristics in supervised setting. For SpanAlign, we quote results from Nagata et al. (2020).

	Dataset	Train	Val	Test
zero-shot cross-lingual transfer (unseen alignment)	Align6	3,362	-	-
	de-en	-	-	508
	ro-en	-	-	248
	fr-en	-	-	447
	zh-en	-	-	450
	ja-en	-	-	582
	sv-en	-	192	-
fully supervised	de-en	300	-	208
	ro-en	150	-	98
	fr-en	300	-	147
	zh-en	450	-	450
	ja-en	653	225	357

Table 8: Number of training, validation and test samples in different settings. We omit few-shot as it shares the same test set as the fully supervised setting but only use 32 samples for training in each language.