WSPAlign: Word Alignment Pre-training via Large-Scale Weakly Supervised Span Prediction

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]{qiyuw, yoshimasa-tsuruoka}@g.ecc.u-tokyo.ac.jp
[2]masaaki.nagata@ntt.com

Abstract

Most existing word alignment methods rely on manual alignment datasets or parallel corpora, which limits their usefulness. Here, to mitigate the dependence on manual data, we broaden the source of supervision by relaxing the requirement for correct, fully-aligned, and parallel sentences. Specifically, we make noisy, partially aligned, and non-parallel paragraphs. We then use such a large-scale weakly-supervised dataset for word alignment pre-training via span prediction. Extensive experiments with various settings empirically demonstrate that our approach, which is named WSPAlign, is an effective and scalable way to pre-train word aligners without manual data. When fine-tuned on standard benchmarks, WSPAlign has set a new state of the art by improving upon the best supervised baseline by **3.3~6.1** points in F1 and **1.5~6.1** points in AER . Furthermore, WSPAlign also achieves competitive performance compared with the corresponding baselines in few-shot, zero-shot and cross-lingual tests, which demonstrates that WSPAlign is potentially more practical for low-resource languages than existing methods. [1]

1 Introduction

Word alignment, which aims to align the corresponding words in parallel texts, is a fundamental Natural Language Processing (NLP) task that was originally developed as an important supporting task for statistical machine translation. While deep end-to-end models have become the mainstream solution for machine translation, word alignment is still of great importance in many NLP scenarios, such as projecting linguistic annotations (David et al., 2001) and XML markups (Hashimoto et al., 2019), post-editing for detecting problem of under-translation (Tu et al., 2016), and enforcing pre-specified terminology constraints in translation (Song et al., 2019). Besides, word alignment can also improve the cross-lingual language pre-training (Chi et al., 2021).

However, most existing word alignment methods rely on either manual alignment datasets or parallel corpora for training, which weakens their usefulness because of the limiting accessibility of data. An additional weakness with requiring manual data is the generalization ability because deep models trained on a dataset can fail on other datasets. Therefore, these existing approaches are also limited in terms of potential cross-lingual use. On the other hand, recent studies (Mahajan et al., 2018; Kolesnikov et al., 2020; Chen et al., 2021; Galvez et al., 2021; Radford et al., 2022) in various fields leverage weak supervision signals in large-scale data available on the web for pre-training, which is a promising alternative to training on manual data.

Inspired by this, we propose to utilize Wikipedia and multi-lingual Pre-trained Language Models (PLMs) to make large-scale word alignment supervision for pre-training via span prediction. We broaden the source of supervision by relaxing the requirements for **correct** (manually made), **fully-aligned** (all words in a sentence pair are annotated), and **parallel** sentences. Specifically, we make **noisy** (automatically made), **partially-aligned**, and **non-parallel** paragraphs (or monolingual paragraph pairs). We make automatic partial alignment between non-parallel sentences from either co-mentions [2] of entities obtained through Wikipedia hyperlinks or alignments of common words based on the similarity of contextual word embeddings.

We name our method WSPAlign, which is short for **W**eakly **S**upervised span **P**rediction pre-training for word **Align**ment. With weak supervision, we are potentially able to scale the pre-training data up to millions of paragraph pairs in

[1]The source code is publicly available at https://github.com/qiyuw/wspalign.

[2]Co-mention means two paragraphs mention an identical entity.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11084–11099

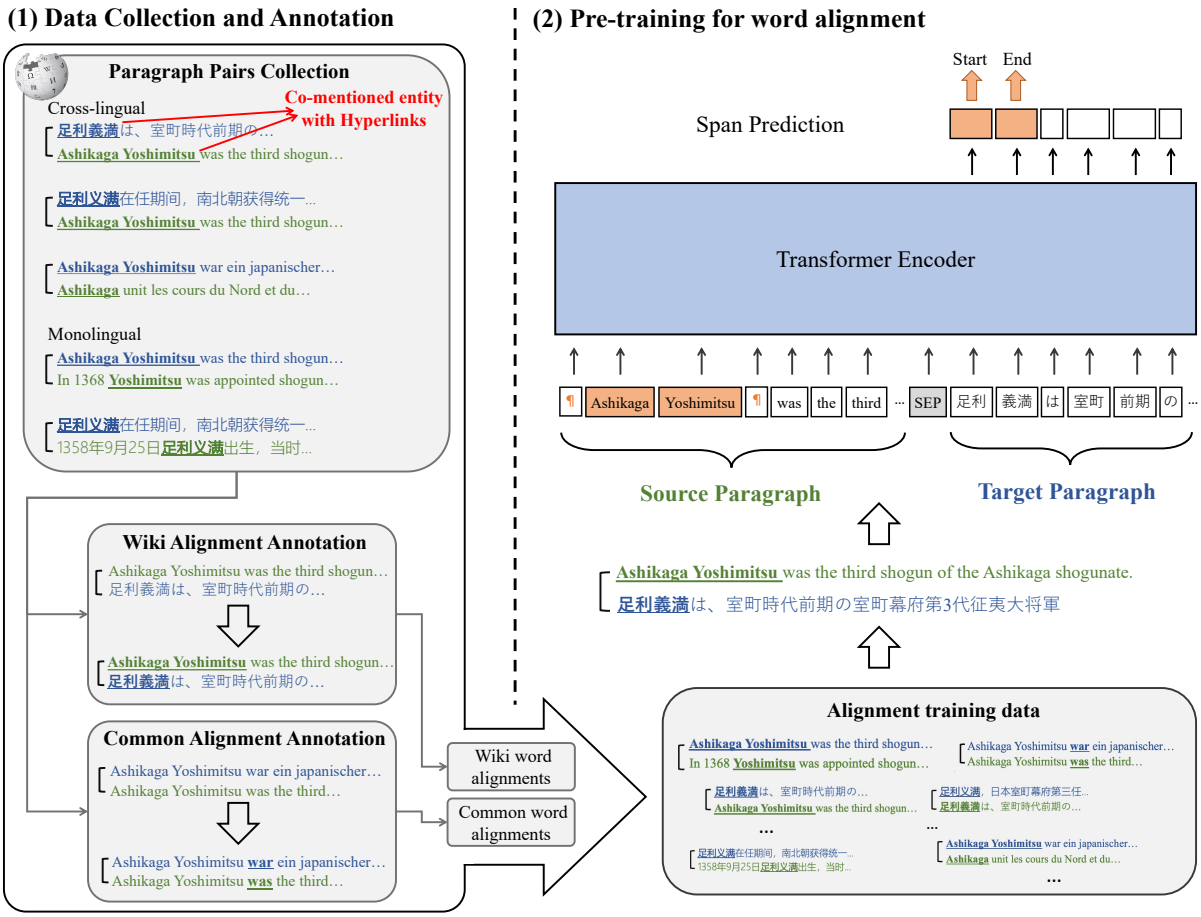


Figure 1: Framework of WSPAlign. Paragraphs are all collected from Wikipedia. We first collect paragraph pairs in which two paragraphs contain an identical language-agnostic entity. Note that the paragraph pairs can be cross-lingual or monolingual, depending on the downstream application goals. Then, we automatically annotate word alignments for common words and wiki words separately and combine them together to make the final dataset. Lastly, the model is pre-trained on the above collected weakly-supervised datasets via the span prediction task.

hundreds of languages. For instance, we made tens of millions of paragraph pairs and sampled a dataset with 2 million pairs in experiments, far more than 5,000 training examples in the existing benchmark dataset. With no requirement for manual datasets, our pre-training approach makes word aligners more practical. Extensive experiments provide empirical evidence for WSPAlign's effectiveness in zero-shot, few-shot and supervised settings. We also conduct monolingual pre-training to test WSPAlign's cross-lingual ability.

2 Related Work

Word Alignment Recent word aligners based on pre-trained language models, such as SimAlign (Jalili Sabet et al., 2020), AWESoME (Dou and Neubig, 2021) and SpanAlign (Nagata et al., 2020; Chousa et al., 2020), have significantly outperformed previous word aligners based on statistical machine translation, such as Giza++ (Och and Ney, 2003) and FastAlign (Dyer et al., 2013). SimAlign is an unsupervised word aligner based on the similarity of contextualized word embeddings. AWESoME and SpanAlign are supervised word aligners that are trained on parallel corpora and manual word alignments, respectively. Particularly, Nagata et al. (2020) proposed to formalize this problem as supervised span prediction using PLMs like BERT (Devlin et al., 2019), which had set the new state of the art on multiple standard benchmarks without the need for parallel corpora. Inspired by this, we take span prediction as our pre-training objective in this paper.

Weakly Supervised Pre-training Recent PLMs in the field of NLP, e.g., GPT-3 (Brown et al., 2020), have shown remarkable zero-shot performance on various tasks without requiring any task-specific datasets. Although understudied so far for word

alignment, recent studies in other fields such as computer vision (Mahajan et al., 2018; Kolesnikov et al., 2020) and speech recognition (Chen et al., 2021; Galvez et al., 2021; Radford et al., 2022) have shown that weakly supervised but larger datasets surpass manual ones with gold labels in terms of robustness and generalization of models. This suggests that large-scale weakly-supervised pre-training is a promising alternative to manually collected supervised datasets or parallel corpora.

3 Appraoch

3.1 Background

We investigate the possibility of word alignment based on span prediction because it is state-of-the-art when manual alignment data is available. Nagata et al. (2020) proposed to frame word alignment as a SQuAD-style span prediction task (Rajpurkar et al., 2016). In SQuAD-style question answering, given a *context* extracted from a Wikipedia paragraph and a *question*, the goal is to predict the answer span within the context based on the given question. Word alignment can be framed similarly, as shown in the top-right part in Figure 1. Given a source sentence with a source token specified by the special token ¶, the goal is to predict the aligned tokens in the target sentence.

Formally, given a source sentence $X = x_1, x_2, ..., x_{|X|}$ consisting of $|X|$ characters, a source token $X_{ij} = x_i...x_j$ that spans (i, j) in the source sentence, a target sentence $Y = y_1, y_2, ..., y_{|Y|}$ consisting of $|Y|$ characters, the objective is to predict the target token $Y_{kl} = y_k, ..., y_l$ that spans (k, l) in the target sentence.

Following the settings in Devlin et al. (2019) for the SQuAD v2.0 task, the target span can be extracted by predicting the start and end position in the target sentence. The probabilities of the start and end positions of the answer span can be defined as p_{start} and p_{end}, respectively. Given the score $w_{ijkl}^{X \to Y}$ as the product of p_{start} and p_{end}, the training objective is to select the answer span $(\hat{k}, \hat{l})$ in the target sentence Y that maximizes the score $w_{ijkl}^{X \to Y}$, based on the source sentence X and source span (i, j), as shown in the following equations,

$$w_{ijkl}^{X \to Y} = p_{start}(k|X, Y, i, j) \times p_{end}(l|X, Y, i, j), \quad (1)$$

$$(\hat{k}, \hat{l}) = \underset{(k,l):1<k<l<|Y|}{\arg\max} w_{ijkl}^{X \to Y}. \quad (2)$$

While span prediction works well on word alignment, it still requires datasets with manually aligned parallel sentences. In the following subsections, we propose to pre-train a word alignment model with a large-scale weakly-supervised dataset.

Algorithm 1: Paragraph Pair Collection from Wikipedia

Input: Multilingual paragraph set $\mathcal{P}$
Language-agnostic entity set $\mathcal{E}$

Initialize an empty paragraph pair list $\mathcal{C}$ and inverted index dictionary $\mathcal{I}$;
foreach *paragraph* $p \in \mathcal{P}$ **do**
  `// Get every entity in` p `by the hyperlink`
  $\mathcal{E}_p := GetEntities(p)$;
  foreach $e \in \mathcal{E}_p$ **do**
    Append p into $\mathcal{I}[e]$;
foreach *entity* $e \in \mathcal{E}$ **do**
  Find all paragraphs that mentioned e, $\mathcal{P}_e \subseteq \mathcal{P} := \mathcal{I}[e]$;
  Make pair-wise combination for $\mathcal{P}_e$ and append to $\mathcal{C}$;

Output: Paragraph Pairs with Co-mentioned Entity $\mathcal{C}$

3.2 Data Collection and Annotation

Figure 1 shows the framework of our proposed approach. Firstly, we collect Wikipedia paragraph pairs by co-mentioned hyperlinks. A typical Wikipedia page contains paragraphs mentioning entities with hyperlinks. A hyperlink points to a language-agnostic entity with a unique entity identifier provided by a public project called Wikidata[3]. We use those identifiers to build an inverted index dictionary, in which each key is an entity identifier and its corresponding value is a list of paragraphs that mention the entity. On the basis of this dictionary, we make two paragraphs as a pair if they are indexed by the same entity, i.e., they contain hyperlinks with the same unique identifier. These two paragraphs can be in any language and on any page. Algorithm 1 elaborates on the collection process.

After obtaining the paragraph pairs, we automatically annotate the word alignments. We categorize words into ***common words*** and long-tailed ***wiki words*** and address them separately.

[3] https://en.wikipedia.org/wiki/Wikipedia:Wikidata

Annotation for Common words Common words can be defined by existing common word lists[4] or part-of-speech (POS) tags. In this paper, we use a POS tagger to identify whether a word has a POS tag for common words or not. The common POS tags we used are shown in §A.2. We take the method in Jalili Sabet et al. (2020), which shows reliable unsupervised ability for word alignment with contextual embeddings in a PLM, to annotate alignments for common words. We make alignments by bi-directional agreement, i.e., two tokens are identified as aligned if they are the most similar token in each other's paragraph. Lastly, we only keep alignments in which at least one of the aligned tokens is common words; otherwise we discard it.

Annotation for wiki words A wiki word here denotes a token span in a paragraph. The token span is associated with a hyperlink pointing to an entity, as we introduced in the data collection. Hence, regardless of what languages in which the wiki words are mentioned, we can make alignments for wiki words by directly aligning the corresponding hyperlinks spans of that co-mentioned entity.

It is necessary to have separate processes for common words and wiki words because wiki words are mainly named entities, we need alignments for common words to complement them. It is known that embedding-based methods work well on annotating common word alignments but perform badly for long-tail wiki words as the embeddings of those long-tail words are usually poorly optimized and noisy (Bahdanau et al., 2017; Gong et al., 2018; Khassanov et al., 2019; Schick and Schütze, 2020; Wu et al., 2021) in a PLM.

The wiki word and common word alignments are denoted as $\mathcal{D}_{wiki}$ and $\mathcal{D}_{com}$, respectively. The formal definition is given in §A.1. After data collection and annotation for wiki words and common words, we combine the two weakly-supervised datasets to obtain the final pre-training dataset, denoted by $\mathcal{D} = \mathcal{D}_{com} \cup \mathcal{D}_{wiki}$.

3.3 Word Alignment Pre-training via Weakly-Supervised Span Prediction

Training Objective We utilize span prediction as our pre-training objective, as shown in Figure 1. As introduced in §3.1, given a alignment example $(X, Y, i, j, k, l) \in \mathcal{D}$, the objective is to optimize a backbone neural network f_{θ^b}, a start position predictor g_{θ^s} and an end position predictor g_{θ^e}, which are parameterized by θ^b, θ^s and θ^e, respectively. The predicted probabilities that (k, l) are the start and end positions of the aligned span in Y can be respectively computed as follows,

$$
\begin{aligned}
prob(t, \theta^1, \theta^2) &= \frac{e^{g_{\theta^2}(f_{\theta^1}(X,Y,i,j))_t}}{\sum_{m=1}^{|Y|} e^{g_{\theta^2}(f_{\theta^1}(X,Y,i,j))_m}} \\
p_{start}(k|X, Y, i, j) &= prob(k, \theta^b, \theta^s) \\
p_{end}(l|X, Y, i, j) &= prob(l, \theta^b, \theta^e)
\end{aligned} \tag{3}
$$

Then the networks can be applied to X, Y and (i, j) to compute the score $w_{ijkl}^{X \to Y}$ based on Equation 1. Following the setting in BERT (Devlin et al., 2019), we optimize $\Theta = \{\theta^b, \theta^s, \theta^e\}$ with the following loss for each training example,

$$
L(X, Y, i, i, k, l; \Theta) = -\log w_{ijkl}^{X \to Y} \tag{4}
$$

Inference and Fine-tuning After the pre-training is finished, the model can be directly used to predict word alignments as follows. Given a source sentence X, source span (i, j) and target sentence Y, the target span $(\hat{k}, \hat{l})$ can be predicted by Equations 1 and 2. This setting is denoted as *zero-shot*. Moreover, our pre-trained model can be easily improved further by fine-tuning on available manual word alignment datasets. Supervised word alignment is viable because a small amount of gold alignment data can be annotated in hours (Stengel-Eskin et al.; Nagata et al., 2020), which is a reasonable budget in practice if we want to make it perform better on a specific low-resource language pair. The settings in which a small number and all training examples are used are denoted as *few-shot* and *supervised* fine-tuning, respectively. The experimental settings of few-shot and supervised fine-tuning are the same, except for an increased number of training epochs performed in the few-shot setting. Details are shown in §4.3.

Mapping Character-based Prediction to Word Tokens As our approach is span-prediction based, the predicted spans may not always align with the original word boundaries. Therefore, following implementation in previous work (Nagata et al., 2020), we select the longest sequence of target tokens that are strictly included in word boundaries in the target sequence as the predicted span. For example, if the model predicts [Yo, ##shi, ##mits, ##u, AS, ##HI], we select [Yo, ##shi, ##mits, ##u] as the predicted span because [AS, ##HI] is not strictly included in a word.

[4]For example, https://www.wordfrequency.info/

Symmetric Word Alignment The model performs a one-way prediction of the aligned span for the given source tokens. Such an asymmetric prediction can result in inconsistent alignments when we swap the source and target. We follow the strategy in SpanAlign (Nagata et al., 2020) to solve it and obtain the final alignment. Specifically, we can first obtain the token-level alignment probabilities predicted by the model separately in two directions for a pair of sentences. Then, we calculate the symmetric probabilities for each token pair by simply averaging the two probability scores. Lastly, we identify two tokens as aligned if the symmetric probability is larger than a preset threshold.

4 Experiments

4.1 Pre-training Dataset Details

We pre-train our model in a weakly-supervised manner, in which all pre-training data are automatically collected and annotated in the way described in §3.2. We first collect paragraphs from Wikipedia dumps[5] in English, German, French, Romanian, Chinese and Japanese. Statistics of paragraphs and entities are shown in Table 5 in Appendix §A. The connections of inter-language hyperlinks are extracted from Wikidata[6]. We use `Wikipedia2Vec` [7](Yamada et al., 2020) to extract the paragraphs and co-mention relations of entities. In this paper, we make the paragraph pairs English-centric, i.e., De-En (German-English), Fr-En (English-Frence), Ro-En (Romanian-English), Zh-En (Chinese-English) and Ja-En (Japanese-English), for more efficient pre-training because most available benchmarks are English-centric. The numbers of sampled examples in each language pair are equal.

Additionally, we also collect a monolingual dataset in English for testing WSPAlign's cross-lingual ability, the experimental analysis of which is shown in §5.1. The collection process of monolingual data is the same as that of multi-lingual data, except for an additional filter for cross-lingual mentioned entities. That is, we keep only the entities that have been mentioned in another language at least once. We did this for two reasons: one is the explosive computational cost for co-mentions within a language, and we also want entities that appear across various languages because we are testing the cross-lingual alignment ability.

Prior to annotating the alignment, we filter those paragraph pairs by length for more stable training. Specifically, we keep only the pairs with medium length , i.e., the pairs that include paragraphs longer than 158 subwords and shorter than 30 subwords are removed. We use SentencePiece with checkpoint `flores101_mm100_615M` [8](Goyal et al., 2022) to tokenize paragraphs in multiple languages, assuming that each sub-word contains a similar amount of information. After that, we further filter the pairs by semantic similarity because a pair with two unrelated paragraphs is likely to result in no aligned common words between them. Hence, we keep only the paragraph pairs with a semantic similarity score higher than 0.75, in which the score is calculated by the cosine distance on the embeddings encoded by recent sentence embedding methods. We use `LaBSE` [9](Feng et al., 2022) and `pcl-bert-base-uncased` [10](Wu et al., 2022) as the sentence embedders for multi-lingual and monolingual datasets, respectively.

Lastly, we randomly sample 2,000,000 pairs as the final dataset. As introduced in §3.2, we annotate wiki word alignments for all the 2,000,000 pairs but annotate only randomly selected 200,000 of them for common word alignments. This is because, on average, a paragraph contains more weakly-supervised alignments for common words than wiki words. We use the POS tagger `flair/upos-multi` [11](Akbik et al., 2019) to identify common words. The statistics in different stages of data collection and annotation are shown in § A.3.

4.2 Benchmark Datasets

We evaluate WSPAlign's performance on five gold word alignment datasets: Chinese-English (Zh-En), Japanese-English (Ja-En), German-English (De-En), Romanian-English (Ro-En) and English-French (En-Fr).

The Zh-En data is obtained from the GALE Chinese-English Parallel Aligned Treebank (Li et al., 2015). We follow Nagata et al. (2020) to pre-process the data, in which we use Chinese character-tokenized bitexts, remove mismatched bitexts and time stamps, etc. Then we randomly split the dataset into 80% for fine-tuning, 10% for

[5] https://dumps.wikimedia.org/
[6] https://en.wikipedia.org/wiki/Help:Interlanguage_links
[7] https://wikipedia2vec.github.io/wikipedia2vec/
[8] https://github.com/flairNLP/flair
[9] https://huggingface.co/sentence-transformers/LaBSE
[10] https://github.com/qiyuw/PeerCL
[11] https://huggingface.co/flair/upos-multi

testing and 10% for future reserves.

The Ja-En data is obtained from the Kyoto Free Translation Task (KFTT)[12] word alignment data (Neubig, 2011). KFTT word alignment data is made by aligning part of the dev and test translation data. We use all eight dev files for fine-tuning, four out of seven test files for testing and the remaining three for future reserves.

The De-En data is from Vilar et al. (2006)[13]. The Ro-En data and En-Fr data are from the shared task of the HLT-NAACL-2003 Workshop on Building and Using Parallel Texts (Mihalcea and Pedersen, 2003), and the En-Fr data is originally from Och and Ney (2003). We use the pre-processing and scoring scripts[14] provided by Zenkel et al. (2019) for the De-En, Ro-En and En-Fr data, and the number of sentences are 508, 248 and 447, respectively. For De-En and En-Fr, We use 300 sentences for fine-tuning and the remaining for testing. For Ro-En, we use 150 sentences for fine-tuning and the remaining for testing.

4.3 Experimental Details

Pre-training Setups We conduct continual pre-training for 100,000 steps with 2,000 warmup steps, starting from multilingual PLMs. We use `bert-base-multilingual-cased` [15] (Devlin et al., 2019) for Zh-En and Ja-En, and `xlm-roberta-base` [16] (Conneau et al., 2020) for De-En, En-Fr and Ro-En, respectively. Detailed discussion regarding the choice of PLMs is in §5.3. We carry out preliminary grid searches on the manual KFTT (Ja-En) training set to decide the hyperparameters. The learning rate is set to 1e-6, the maximum sequence length is set to 384, and the batch size is 96. We use a 12-layer Transformer as the encoder, in which the hidden size is 768, and the number of attention heads is 12.

Fine-tuning Setups For testing the performance on downstream datasets, we fine-tuned the pre-trained model for five epochs for *supervised* and 250 epochs for *few-shot* setting, respectively. The labeled examples we use for *few-shot* is 32. Following the common practices of pre-training methods, the hyperparameters of fine-tuning are decided empirically by grid-search on the development set. Learning rate is selected from {1e-6, 3e-6, 1e-5, 3e-5} and batch size is selected from {5, 8, 12}. Besides, the threshold for symmetric word alignment described in §3.3 is set to 0.4, following SpanAlign (Nagata et al., 2020).

[12] http://www.phontron.com/kftt

[13] https://www-i6.informatik.rwth-aachen.de/goldAlignment/

[14] https://github.com/lilt/alignment-scripts

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

[16] https://huggingface.co/xlm-roberta-base

4.4 Measures for Word Alignment Quality

We measure word alignment quality by precision, recall and F1 score in the same way as previous literature (Nagata et al., 2020). Given the predicted alignment results (A), *sure* alignments (S) and *possible* alignments (P). Precision, Recall, and F1 can be calculated as:

$$\begin{aligned} Precision(A,P) &= \frac{|A \cap P|}{|A|} \\ Recall(A,S) &= \frac{|A \cap S|}{|S|} \\ F_1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \end{aligned} \tag{5}$$

We also report Alignment Error Rate (AER) (Och and Ney, 2003), which can be calculated as equation 6, but regard it as a secondary metric because we take the previous literature's (Fraser and Marcu, 2007; Nagata et al., 2020) claim that AER inappropriately favors precision over recall and should be used sparingly.

$$AER(A,S,P) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{6}$$

Note that only partial word alignment datasets (in our paper, De-En and En-Fr) may distinguish between sure and possible alignments. In the case where *possible* and *sure* alignments are not distinguished (i.e., P == S), AER = 1 - F1. We report both because previous work calculates and reports results in different ways. In particular, as the En-Fr dataset is known as noisy, special handling was necessary for evaluation in previous studies. And the reported F1 numbers in previous baselines vary greatly due to the different evaluation methods. Consequently, we choose a common practice that fine-tuning on the *sure* data but evaluating on the *sure+possible* data, and we only report AER for En-Fr for a fairer comparison.

4.5 Main Quantitative Results

In this section, we use all available training examples in the benchmark datasets to reach the best potential of WSPAlign in the *supervised* fine-tuning setting. The competitive baselines include Giza++,

Test Set	Method	Precision	Recall	F1	AER
Zh-En	FastAlign (Stengel-Eskin et al.)	80.5	50.5	62.0	-
	DiscAlign (Stengel-Eskin et al.)	72.9	74.0	73.4	-
	SpanAlign (Nagata et al., 2020)	84.4	89.2	86.7	13.3
	WSPAlign (ours)	90.8	92.2	**91.5** (↑ **4.8**)	**8.5** (↓ **4.8**)
Ja-En	Giza++ (Neubig, 2011)	59.5	55.6	57.6	42.4
	AWESoME (Dou and Neubig, 2021)	-	-	-	37.4
	SpanAlign (Nagata et al., 2020)	77.3	78.0	77.6	22.4
	WSPAlign (ours)	81.6	85.9	**83.7** (↑ **6.1**)	**16.3** (↓ **6.1**)
De-En	SimAlign (Jalili Sabet et al., 2020)	-	-	81.0	19.0
	AWESoME (Dou and Neubig, 2021)	-	-	-	15.0
	SpanAlign (Nagata et al., 2020)	89.9	81.7	85.6	14.4
	WSPAlign (ours)	90.7	87.1	**88.9** (↑ **3.3**)	**11.1** (↓ **3.3**)
Ro-En	SimAlign (Jalili Sabet et al., 2020)	-	-	71.0	29.0
	AWESoME (Dou and Neubig, 2021)	-	-	-	20.8
	SpanAlign (Nagata et al., 2020)	90.4	85.3	86.7	12.2
	WSPAlign (ours)	92.0	90.9	**91.4** (↑ **4.7**)	**8.6** (↓ **3.6**)
En-Fr	SimAlign (Jalili Sabet et al., 2020)	-	-	93.0	7.0
	AWESoME (Dou and Neubig, 2021)	-	-	-	4.1
	SpanAlign (Nagata et al., 2020)	97.7	93.9	-	4.0
	WSPAlign (ours)	98.8	96.0	-	**2.5** (↓ **1.5**)

Table 1: Comparison of WSPAlign and previous methods on word alignment datasets. Higher F1 scores are better. Lower AER scores are better. We highlight the best number in the same setting and test set with bold font.

SimAlign, AWESoME and SpanAlign, whose details are described in §2. For baselines, we report the best numbers in their original paper.

Table 1 shows the comparison of our proposed model and existing approaches. It demonstrates that WSPAlign significantly outperforms all supervised and unsupervised baselines. Specifically, WSPAlign improves the best supervised baseline by **3.3~6.1** points in F1 and **1.5~6.1** points in AER.

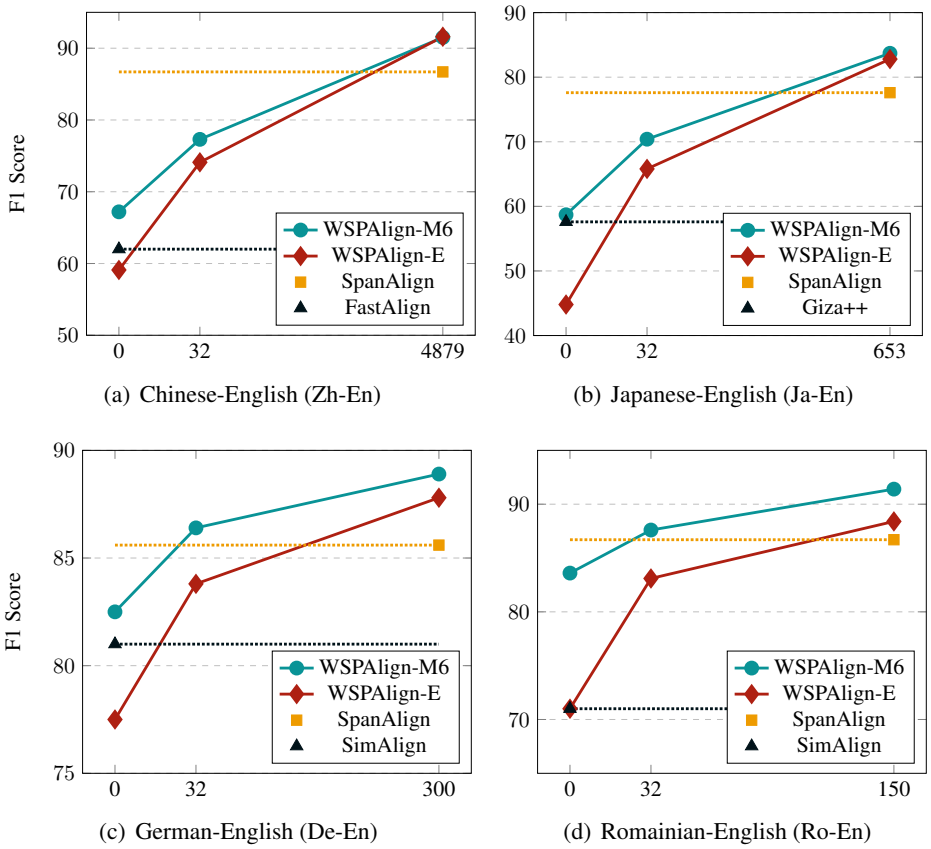
Additionally, we can observe that WSPAlign improves the baselines on Ja-En with a relatively larger margin. As Japanese is known as a language distant from English, this indicates WSPAlign's superiority in word alignment in difficult language pairs by introducing more cross-lingual information in the pre-training.

4.6 Zero-shot and Few-shot Performance

With varying scales of manual training examples used after the pre-training, we evaluate the zero-shot and few-shot performance of WSPAlign. As shown in Figure 2, we test 0 (*zero-shot*), 32 (*few-shot*) and the full amount (*supervised*) of examples in the benchmark datasets. Details regarding the implementation can be found in §3.3 and §4.3.

The circle points with the green line show the performance trend of WSPAlign pre-trained on weakly supervised data in six languages (WSPAlign-M6 in Figure 2). For all test sets, zero-shot WSPAlign-M6 outperforms the unsupervised baselines, and the few-shot WSPAlign-M6 with only 32 training examples significantly outperforms the unsupervised baselines by a large margin. This indicates that the proposed pre-training method has a basic zero-shot word alignment ability with no need for any manual data, and the performance can be further improved with only a small number of training examples.

Notably, zero-shot WSPAlign-M6 beats the unsupervised baselines by a large margin and almost reaches the performance of the supervised baseline on Ro-En. On Ro-En and De-En, WSPAlign-M6 even slightly outperforms the fully supervised baseline. As English is known to be closer to Romanian and German than Chinese and Japanese, the results imply that the proposed approach has a higher reward when the downstream languages to be aligned are close. Additionally, the Ro-En and De-Rn datasets respectively include only 150 and 300 training examples, which can make the supervised methods not perform satisfactorily. Thus, considering the computation cost of the pre-training in practice, our proposed large-scale span prediction pre-training with weakly supervised data can bring more benefits in the case when avail-



(a) Chinese-English (Zh-En) (b) Japanese-English (Ja-En)

(c) German-English (De-En) (d) Romainian-English (Ro-En)

Figure 2: Comparison of varying scales of manual samples used on four word alignment test sets. The y-axis is F1 score and the x-axis is the number of manual samples used in the training.

able manual data are scarce or the downstream languages are close.

5 Discussion

5.1 Mono-lingual Span Prediction Pretraining

In this section, we will examine mono-lingual span prediction pretraining by pre-training on English-only data but testing on other languages, to investigate the potential cross-lingual ability of WSPAlign to confirm whether it is ready for practical application. Although Wikipedia and recent multilingual PLMs support hundreds of languages, the amount of information available for minority languages can still be small [17]. How to address such language equality problems is often discussed in recent NLP research (Conneau et al., 2020; Costa-jussà et al., 2022). In the scope of this paper, even if we collect supervision signals from large-scale encyclopedias and PLMs, the datasets could still be limited for exceptionally low-resource languages in practice.

The diamond points with the red line in Figure 2 show the performance trend of WSPAlign pretrained on English-only alignment data, i.e., WSPAlign-E. We observe that zero-shot WSPAlign-E underperforms the unsupervised baseline, except on the easier Ro-En test set. However, WSPAlign-E can be significantly improved and outperforms the existing unsupervised baselines with only 32 manual examples, which can be collected at a low cost. If we further fine-tune WSPAlign-E with a full supervised dataset, it can outperform the supervised baseline on all test sets. These observations show that with only pre-training on monolingual weakly supervised alignments, WSPAlign is not able to be a better word aligner than the existing ones, although it achieves a basic zero-shot ability. However, fine-tuning it on a small number of manual examples can be a practical cross-lingual word aligner better than unsupervised baselines. Moreover, it can beat the state-of-the-art method when the same amount of manual examples are available.

Such a cross-lingual transferring ability that holds for zero-shot, few-shot, and supervised set-

[17] https://meta.wikimedia.org/wiki/List_of_Wikipedias

	P	R	F1	AER
SpanAlign	84.4	89.2	86.7	13.3
WSPAlign	90.8	92.2	91.5	8.5
w/o common words	91.3	85.4	88.3	11.7
w/o Wiki words	91.5	86.0	88.6	11.4

Table 2: Ablation study by removing common words or wiki items for alignment. Performance on Zh-En test set. Higher F1 is better and lower AER is better.

Test Set	mBERT				XLM-R			
	P	R	F1	AER	P	R	F1	AER
Zh-En	90.8	92.2	91.5	8.5	83.6	91.4	87.3	12.7
Ja-En	81.6	85.9	83.7	16.3	81.2	83.8	82.5	17.5
De-En	91.9	84.9	88.3	11.7	90.7	87.1	88.9	11.1
Ro-En	89.6	89.5	89.5	10.5	92	90.9	91.4	8.6

Table 3: The performance on the test sets with different PLM initiation. We highlight the better performance in the same setting with the yellow box.

tings suggests that WSPAlign is potentially very practical for low-resource languages by only pre-training on large-scale monolingual data, as low-resource language resources are always hard to collect.

From another perspective, our proposed WSPAlign consists of two components: span prediction and bilingual equivalence identification. As an ablation study of WSPAlign, mono-lingual span prediction pre-training performs without bilingual equivalence knowledge but only learns the span prediction. Intriguingly, mono-lingual span prediction still improves bilingual word alignment accuracy in the above experiments. A possible explanation for this result is that word embeddings are somehow aligned out of the box in a multilingual language model. This indicates that only optimizing on mono-lingual span prediction in our proposed method can also potentially generalize to cross-lingual word alignment.

5.2 Effect of Common Words and Wiki Words

We test two variants of WSPAlign by removing the common words and wiki words in the pre-training data, i.e., WSPAlign w/o common and WSPAlign w/o Wiki. We chose the largest benchmark dataset Zh-En and the setting of supervised fine-tuning for testing. The experimental results in Table 2 show that when alignments for common words or wiki words are removed from the training data, the performance of WSPAlign will drop by about 3 points on F1 and AER. But both two variants outperform the supervised baseline SpanAlign. This indicates that the improvement from our proposed weakly supervised pre-training still holds even when we make alignments only for either common words or wiki words, and using both leads to better performance.

5.3 The Choice of Multi-lingual PLMs

Besides the span prediction pre-training we propose, WSPAlign still needs a prior conventional language pre-training to ensure the basic ability of language understanding. In this paper, we start the span prediction pre-training from the checkpoint of two popular multi-lingual PLMs, mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). To investigate the effect of different PLMs used, we compare the performance of WSPAlign with mBERT and XLM-R on all test sets except En-Fr. We do not use En-Fr because the dataset is noisy. Table 3 clearly shows that mBERT performs better on Zh-En and Ja-En. In contrast, XLM-R performs better on De-En and Ro-En. Such a difference in performance may be caused by the tokenization method during the language pre-training. The byte-level sub-word tokenization used in RoBERTa (Liu et al., 2019) can work poorly for Chinese and Japanese because the character is the smallest unit in these languages. Hence we use mBERT as the initialization checkpoint for Zh-En and Ja-En and XLM-R for the rest. We also suggest choosing the appropriate PLM for WSPAlign according to the downstream languages in practice.

6 Conclusion

In this paper, we propose to pre-train word aligners with weakly-supervised signals that can be automatically collected. We broaden the source of supervision by relaxing the requirement for correct, fully-aligned, and parallel sentences. Specifically, we make noisy, partially aligned, and non-parallel paragraphs on a large scale. Experimental results in this paper show that pre-training with large-scale weakly-supervision can significantly improve existing word alignment methods and make word aligners more practical as well because no manual data is needed. We provide empirical evidence of how much large-scale span prediction pre-training can help word alignment in terms of data accessibility, the number of manual examples used, and cross-lingual ability. We hope this paper can contribute to further exploiting practical word alignment techniques with large-scale weak supervision.

Limitations

Although WSPAlign successfully outperforms all existing baselines, it is still limited to the accessibility of low-resource language information. For example, the collection of pre-training data requires multi-lingual POS tagging tools to identify which words are common or not. It also requires a multilingual PLM and Wikipedia hyperlinks to make the alignments, which could be inaccessible for an exceptional minority language. But note that we showed WSPAlign's cross-lingual ability in §5.1, which implies that this issue can potentially be addressed in the direction of pre-training on large-scale monolingual data with our future effort. Besides, this paper lacks evaluation on real low-resource language benchmarks because there is no existing test set. We will try to collect and annotate low-resource word alignment data in our future work.

Ethics Statement

This paper investigates the pre-training for word alignment, which will not lead to a negative social impact. The data used in this paper are all publicly available and are widely adopted in previous literature, avoiding any copyright concerns. The proposed method does not introduce ethical bias. On the contrary, our aim is to advance word alignment techniques to enhance their utility for low-resource language communities, promoting inclusivity and equitable access to language resources.

Acknowledgement

We thank Ryokan Ri for the valuable discussion and assistance with Wikipedia2vec. Qiyu Wu was supported by JST SPRING, Grant Number JPMJSP2108.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Dzmitry Bahdanau, Tom Bosc, Stanislaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *arXiv preprint arXiv:1706.00286*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, He-Yan 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.

Katsuki Chousa, Masaaki Nagata, and Masaaki Nishino. 2020. SpanAlign: Sentence alignment method based on cross-language span prediction and ILP. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard 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.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Yarowsky David, Ngai Grace, Wicentowski Richard, et al. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8.

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.

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.

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.

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.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.

Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. 2021. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. *arXiv preprint arXiv:2111.09344*.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In *Advances in neural information processing systems*, pages 1334–1345.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

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.

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.

Yerbolat Khassanov, Zhiping Zeng, Van Tung Pham, Haihua Xu, and Eng Siong Chng. 2019. Enriching rare word representations in neural language models by embedding matrix augmentation. *arXiv preprint arXiv:1904.03799*.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer.

Xuansong Li, Stephen Grimes, Stephanie Strassel, Xiaoyi Ma, Nianwen Xue, Mitch Marcus, and Ann Taylor. 2015. Gale chinese-english parallel aligned treebank–training.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision*, pages 185–201. Springer.

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.

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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *AAAI*, pages 8766–8774.

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.

Elias Stengel-Eskin, Tzu-Ray Su, Matt Post, and Benjamin Van Durme. A discriminative neural model for cross-lingual word alignment. 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 910–920.

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.

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*.

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. *arXiv preprint arXiv:2201.12093*.

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*.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

A Appendix

A.1 Formal Definition of Annotation for Alignments

Wiki Words Given a paragraph pair X and Y, X and Y contain an identical entity e. Suppose (i, j) and (k, l) are the spans [18] of e in X and Y respectively, we add the alignment of (X_{ij}, Y_{kl}) into the dataset $\mathcal{D}_{wiki}$.

Common Words Assume we have a parameterized network δ (e.g., a PLM) that can be applied to a token X_{ij} in the paragraph to derive a dense real-valued vector $\mathbf{h}_{ij}^X = \delta(X_{ij}) \in \mathbb{R}^d$. Then we can calculate the similarity scores for the embedding of each token in the source paragraph X and target paragraph Y, and obtain pairwise similarity scores S for every token in the paragraph pairs, $S_{ijkl}^{X \to Y} = sim(\mathbf{h}_{ij}^X, \mathbf{h}_{kl}^Y)$, where sim is a similarity function for two vectors, e.g., cosine similarity. Then, for two words (i, j) in source sentence X and (k, l) in target sentence Y, we annotate the alignment of (X_{ij}, Y_{kl}) if and only if $((i, j) = \underset{(i,j):1<i<j<|X|}{\arg\max} S_{ijkl}^{X \to Y}) \wedge ((k, l) = \underset{(k,l):1<k<l<|Y|}{\arg\max} S_{klij}^{Y \to X})$. As we mentioned earlier, embedding-based methods can perform badly on rare words. Thus we further filter out alignments with common words. That is, given an annotated alignment (X_{ij}, Y_{kl}), we add it into the dataset $\mathcal{D}_{com}$ if (i, j) or (k, l) is a common word. Otherwise, we discard it.

[18] The explicit text of the spans can be different, but they refer to the same entity.

POS Tag	Meaning
ADJ	adjective
VERB	verb
DET	determiner
ADP	adposition
AUX	auxiliary
PRON	pronoun
PART	particle
SCONJ	subordinating conjunction
NUM	numeral
NOUN	noun
ADV	adverb
CCONJ	coordinating conjunction
INTJ	interjection

Table 4: The Meaning of POS tags.

	# of entities	# of paragraphs
Zh	1,768,012	22,409,574
En	8,675,433	145,441,685
Ja	1,663,517	51,377,620
Ro	754,005	7,105,064
De	3418485	57,121,818
Fr	3507481	63,551,555

Table 5: Statistics of Wikipedia raw data.

# of paragraph pairs		
	Multi-lingual	Monolingual
with co-mention	89,973,019	72,677,385
– filter by length	41,418,902	40,759,166
– filter by similarity	10,016,210	11,304,002
– finally used	2,000,000	2,000,000
# of alignment annotations		
	Multi-lingual	Monolingual
wiki items	2,000,000	2,000,000
common words*	1,644,019	2,591,357

Table 6: Statistics of paragraph pairs and alignments in the data collection and annotation. *We use only 200,000 pairs for common word alignment.

A.2 Common POS Tags

We use tags shown in Table 4 as common tags [19]. Tokens predicted as one of these tags are identified as common words in our method.

A.3 Statistics of datasets

Table 5 shows the statistics of Wikipedia raw data we use. English has the most numbers of paragraphs and entities, while Romanian has the least paragraphs and entities. Besides, we also count the number of paragraph pairs and alignment annotations in different phases while obtaining the pre-training data. Specific statistics is shown in the Table 6.

A.4 Experimental Enviroments

Table 7 shows the experimental environments and training hours in different settings. We used two NVIDIA Tesla A100 (80G) to conduct the pre-training. The pre-training time is around 40 hours. We used Titan X (12G) to conduct the few-shot and supervised fine-tuning, which can be finished in hours for each run. Note that the few-shot fine-tuning has fewer examples but performs 250 epochs, while supervised fine-tuning only performs for 5 epochs.

[19] https://huggingface.co/flair/upos-multi

Setting	GPU	Dataset	# of Training Examples	Training Time (hours)
Pre-training	NVIDIA Tesla A100 (80G)	6 languages	2,000,000	40
		English only	2,000,000	42
Supervised Fine-tuning	NVIDIA Titan Xp (12G)	Zh-En	4,879	6
		Ja-En	653	3
		De-En	300	1
		Ro-En	150	0.25
		En-Fr	300	1
Few-Shot Fine-tuning	NVIDIA Titan Xp (12G)	-*	32	2

Table 7: Experimental environments and training time.* Training time for each dataset in the few-shot setting is approximately equal.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The 'Limitation' section.

☑ A2. Did you discuss any potential risks of your work?
5

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3,4

☑ B1. Did you cite the creators of artifacts you used?
3,4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All artifacts used are freely public

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3,4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Data we use is public.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4

C ☑ Did you run computational experiments?

4,5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4,appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4,appendix

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.