# A Supervised Word Alignment Method based on Cross-Language Span Prediction using Multilingual BERT

**Masaaki Nagata** and **Katsuki Chousa** and **Masaaki Nishino**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{masaaki.nagata.et,katsuki.chousa.bg,masaaki.nishino.uh}@hco.ntt.co.jp

## Abstract

We present a novel supervised word alignment method based on cross-language span prediction. We first formalize a word alignment problem as a collection of independent predictions from a token in the source sentence to a span in the target sentence. Since this step is equivalent to a SQuAD v2.0 style question answering task, we solve it using the multilingual BERT, which is fine-tuned on manually created gold word alignment data. It is nontrivial to obtain accurate alignment from a set of independently predicted spans. We greatly improved the word alignment accuracy by adding to the question the source token's context and symmetrizing two directional predictions. In experiments using five word alignment datasets from among Chinese, Japanese, German, Romanian, French, and English, we show that our proposed method significantly outperformed previous supervised and unsupervised word alignment methods without any bitexts for pretraining. For example, we achieved 86.7 F1 score for the Chinese-English data, which is 13.3 points higher than the previous state-of-the-art supervised method.[1]

## 1 Introduction

Over the last several years, machine translation accuracy has been greatly improved by neural networks (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017). However, word alignment tools, which were developed during the age of statistical machine translation (Brown et al., 1993; Koehn et al., 2007) such as GIZA++ (Och and Ney, 2003), MGIZA (Gao and Vogel, 2008) and FastAlign (Dyer et al., 2013), remain widely used because the improvement of word alignment accuracy has become stagnant.

This situation is unfortunate because word alignment could be used for many downstream tasks including projecting linguistic annotation (Yarowsky et al., 2001), projecting XML markups (Hashimoto et al., 2019), and enforcing terminology constraints (pre-specified translation) (Song et al., 2019). We could also use it for the user interfaces of post-editing to detect such problems as under-translation (Tu et al., 2016).

Word alignment has a long research history. Here, we focus on approaches that use neural networks because they are the state-of-the art. Most previous works that use them for word alignment (Yang et al., 2013; Tamura et al., 2014; Legrand et al., 2016) achieved accuracies that are basically comparable to GIZA++. However, the accuracy of recent works (Garg et al., 2019; Stengel-Eskin et al., 2019; Zenkel et al., 2020) based on the Transformer (Vaswani et al., 2017), which is the state-of-the art neural machine translation model, have started to outperform GIZA++.

Garg et al. (2019) made the attention of the Transformer more closely resembled the word alignment, and achieved better accuracy than GIZA++ when they used alignments obtained from it for supervision. Zenkel et al. (2020) added an alignment layer using a full target context on top of the Transformer and trained it with a loss function that encouraged contiguous alignment and bidirectional agreement. They outperformed GIZA++ without GIZA++ output for supervision. Stengel-Eskin et al. (2019) proposed a supervised word alignment method using the hidden states of the Transformer, and significantly outperformed FastAlign (11-27 F1 points) using a small number of gold word alignments (1.7K-5K sentences). However, both Garg et al. (2019) and Stengel-Eskin et al. (2019) required more than a

[1] Our implementation is available at https://github.com/nttcslab-nlp/word_align/



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 555–565,
November 16–20, 2020. 

million parallel sentences to pretrain their models. Applying these methods to low-resource language pairs and domains is difficult.

In this paper, we present a novel supervised word alignment method that requires no parallel sentences for pretraining and can be trained from fewer gold word alignments (150-300 sentences). It formalizes word alignment as a collection of SQuAD-style span prediction problems (Rajpurkar et al., 2016) and solves them with multilingual BERT (Devlin et al., 2019). We experimentally show that our proposed model significantly outperformed both (Garg et al., 2019) and (Stengel-Eskin et al., 2019).

Our main contribution is that we make supervised word alignment more practical. Stengel-Eskin et al. (2019) argued that supervised word alignment is a viable option. They concluded that alignment annotation could be performed rapidly 4.4 sentences per minute by annotators with minimal experience using a web-based crowd-sourcing interface. Assuming that a small amount of gold word alignment data, which can be annotated in a couple of hours, our proposed method could be used on 104 languages supported by the multilingual BERT.

## 2 Proposed Method

### 2.1 Word Alignment as Question Answering

Figure 1 shows an example of Japanese-English word alignment data and Figure 2 is its illustration. It consists of a token sequence of the L1 language (Japanese), a token sequence of the L2 language (English), a sequence of the aligned token pairs, the original L1 sentence, and the original L2 sentence. For example, the first item of the third line "0-1" represents that the first token "足利" of the L1 sentence is aligned to the second token "ashikaga" of the L2 sentence. The index of the tokens starts from zero.

In this paper, we frame word alignment as a cross-language span prediction problem similar to the SQuAD-style question answering task (Rajpurkar et al., 2016). In SQuAD, given a context (a paragraph from Wikipedia) and a question, the question answering system predicts an answer as a span in the context. Similarly, given a target sentence as the context and a source word as a question, the word alignment system predicts a translation of the source word as the answer, which is a span in the target sentence.

Figure 3 shows an example of converting word alignment data to a SQuAD-style span prediction. In its upper half, the L1 (Japanese) sentence is given as the context. A token in the L2 (English) sentence "was" is given as a question whose answer is span "である" in the L1 sentence. It corresponds to the three aligned token pairs "24-2 25-2 26-2" in the third line of Figure 1.

We can convert the word alignments for a sentence to a set of queries from a token in the L1 sentence to a span in the L2 sentence and a set of queries from a token in the L2 sentence to a span in the L1 sentence. If a token is aligned to multiple spans, we treat it as a question with multiple answers. If a token has no alignment, we treat it as a question without answers.

We call the question's language the source language and the context's language (and the answer's language) the target language. In Figure 3, the source language is English and the target language is Japanese. This is an English-to-Japanese query.

Suppose the question is such a high-frequency word as "of", which might be found many times in the source sentence. We might easily experience difficulty finding the corresponding span in the target sentence without the source token's context.

The lower half of Figure 3 shows two examples of a question with the source token's context. In question_2, the two preceding words "Yoshimitsu ASHIKAGA" and two following words "the 3rd" are attached to the source token "was" with '¶' (pilcrow: paragraph mark) as a boundary marker[2]. As shown in the experiment, the longer the context is, the better the result. We used the whole source sentence as a context, as shown in question_3.

Since there are many null alignments in word alignment, we adopted the SQuAD v2.0 format (Rajpurkar et al., 2018), which explicitly defines cases when there are no answer spans to the question in the given context. For converting word alignment data to SQuAD-style question answering, both the question and the context are taken from the original sentences, not the tokenized sequences. The start and end positions of the answer span are indexes to the character position of the original target sentence. Since each dataset has a

[2]We used '¶' as a boundary marker because it belongs to the Unicode character category "punctuation" and is included in the multilingual BERT vocabulary. It rarely appears in ordinary text.

```
足利 義満 （ あしかが よしみつ ） は 室町 幕府 の 第 3 代 征夷 大 将軍 （ 在位 1368 年 - 1394 年 ）
で あ る 。
yoshimitsu ashikaga was the 3rd seii taishogun of the muromachi shogunate and
reigned from 1368 to1394 .
0-1 1-0 3-1 4-0 7-9 8-10 9-7 10-3 11-4 12-4 13-5 14-6 15-6 17-12 18-14 19-14 21-15
22-15 24-2 25-2 26-2 27-16
足利義満（あしかがよしみつ）は室町幕府の第 3 代征夷大将軍（在位 1368 年-1394 年）である。
Yoshimitsu ASHIKAGA was the 3rd Seii Taishogun of the Muromachi Shogunate and
reigned from 1368 to1394.
```

Figure 1: Word alignment data between Japanese and English ('to1394' is copied as is).

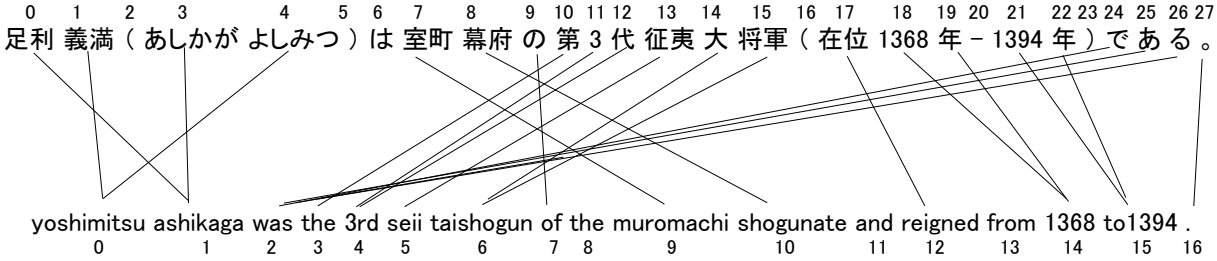


Figure 2: Illustration of the word alignment data. Annotation in Japanese ' あしかがよしみつ' enclosed in the first pair of parentheses is the reading of Chinese characters ' 足利義満' (ASHIKAGA, Yoshimitsu). Annotation in Japanese ' 在位 1368 年-1394 年' enclosed in the second pair of parentheses is plainly translated in English as 'and reigned from 1368 to 1394'.

different standard for tokenization and casing, we only used the tokenization of the source sentence to create a source span in a question.

### 2.2 Cross-Language Span Prediction using Multilingual BERT

We defined our cross-language span prediction task as follows. Suppose we have a source sentence with $|X|$ characters $X = x_1 x_2 \dots x_{|X|}$, and a target sentence with $|Y|$ characters $Y = y_1 y_2 \dots y_{|Y|}$. Given source token $x_{i:j} = x_i \dots x_j$ that covers $(i, j)$ in source sentence $X$, the task is to extract target span $y_{k:l} = y_k \dots y_l$ that covers $(k, l)$ in target sentence $Y$.

We applied multilingual BERT (Devlin et al., 2019) to this task. Although it is designed for such monolingual language understanding tasks as question answering and natural language inference, it works surprisingly well for cross-language span prediction.

For the SQuAD v2.0 task, we used a model described in (Devlin et al., 2019) that added two independent output layers to the pretrained BERT to predict the start and end positions in the context. Suppose $p_{start}$ and $p_{end}$ are the probabilities that each position in the target sentence is the start and end positions of the answer span. We defined score $\omega_{ijkl}^{X \to Y}$ of target span $y_{k:l}$ given source span $x_{i:j}$ as the product of its start and end position probabilities and selected span $(\hat{k}, \hat{l})$ that maximizes $\omega_{ijkl}^{X \to Y}$ as the best answer span:

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

$$(\hat{k}, \hat{l}) = \arg \max_{(k,l):1 \le k \le l \le |Y|} \omega_{ijkl}^{X \to Y} \quad (2)$$

In the SQuAD model of BERT, first, the question and the context are concatenated to generate sequence "[CLS] question [SEP] context [SEP]" as input, where [CLS] and [SEP] are the classification and separator tokens, respectively. Then, the start and end positions are predicted as indexes to the sequence. In the SQuAD v2.0 model, the start and end positions are the indexes to the [CLS] token if there are no answers.

Unfortunately, since the original implementation of the SQuAD model only outputs an answer string, we extended it to output the answer's start and end positions. Inside BERT, the input sequence is first tokenized by WordPiece. It then splits the CJK characters into a sequence of a single character. Since the start and end positions are indexes to the BERT tokens, we converted them to character indexes to make the input tokenization (word boundary) independent of the BERT tokenization.

Figure 4 shows an example of span prediction where the source token is "Yoshimitsu", which consists of four BERT tokens. The original token (word) boundaries are shown by dotted lines.

```
context: "足利義満（あしかがよしみつ）は室町幕府の第 3 代征夷大将軍（在位 1368 年–1394 年）である。"
question_1: "was"
answer: "である",

question_2: "Yoshimitsu ASHIKAGA ¶ was ¶ the 3rd"
question_3: "Yoshimitsu ASHIKAGA ¶ was ¶ the 3rd Seii Taishogun of the Muromachi
Shogunate and reigned from 1368 to1394.
```

Figure 3: English-to-Japanese query without source context (question_1), with limited source context (question_2), and with full source context (question_3)

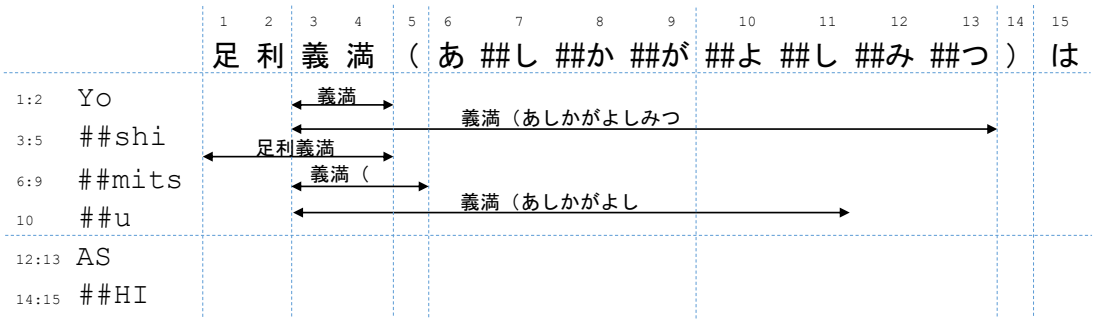


Figure 4: An example of Japanese-to-English span prediction where source token is “Yoshimitsu”. Each BERT token is shown with its character index to the original sentence, and word boundaries are shown by dotted lines.

There are five target span candidates, where “義満” is the correct answer. The predicted target spans do not necessarily agree with the target token boundaries because BERT predicts spans based on its tokens. For target spans that do not agree with the target token boundaries such as “義満（あしかがよし”, we select the longest sequence of the target tokens that is strictly included in the predicted target span such as “義満”, “（”, and “あしかが”, as a set of aligned target tokens from the source token.

### 2.3 Symmetrization of Word Alignments

Since the proposed span prediction model predicts a target span for a source token, it is asymmetric like the IBM model (Brown et al., 1993). To make the span predictions more reliable, we designed a simple heuristics to symmetrize the span predictions of two directions.

Symmetrizing IBM model alignments was first proposed by Och and Ney (2003). One of the most popular Statistical Machine Translation Toolkits, Moses (Koehn et al., 2007), supports a variety of symmetrization heuristics, such as intersection and union, where grow-diag-final is the default. The intersection of the two yields an alignment that consists of one-to-one alignments with higher precision and lower recall than either one separately. The union yields higher recall and lower precision.

As a symmetrization method, for an alignment we averaged the probabilities of the best spans for each token for each direction. A token is aligned if it is completely included in the predicted span. We then extracted the alignments with the average probabilities that exceed a threshold.

Let $x_{i:j}$ be a substring of sentence X that spans (i,j), and let $y_{k:l}$ be a substring of sentence Y that spans (k,l). Let $\omega_{ijkl}^{X \to Y}$ be the probability that token $x_{i:j}$ predicts span $y_{k:l}$, and let $\omega_{ijkl}^{Y \to X}$ be the probability that token $y_{k:l}$ predicts span $x_{i:j}$. Let $\omega_{ijkl}$ be the probability of alignment $a_{ijkl}$ where token $x_{i:j}$ is aligned to token $y_{k:l}$. We define $\omega_{ijkl}$ as the average of probability $\omega_{ij\hat{k}\hat{l}}^{X \to Y}$ of the best predicted span $y_{\hat{k}:\hat{l}}$ from $x_{i:j}$ and probability $\omega_{\hat{i}\hat{j}kl}^{Y \to X}$ of the best predicted span $x_{\hat{i}:\hat{j}}$ from $y_{k:l}$ :

$$\omega_{ijkl} = 1/2(I_{\hat{k} \le k \le l \le \hat{l}}(\omega_{ij\hat{k}\hat{l}}^{X \to Y}) + I_{\hat{i} \le i \le j \le \hat{j}}(\omega_{\hat{i}\hat{j}kl}^{Y \to X})) \quad (3)$$

where $I_A(x)$ is an indicator function. $I_A(x)$ returns $x$ if $A$ is true and 0 otherwise. We regard $x_{i:j}$ and $y_{k:l}$ as aligned if $\omega_{ijkl}$ is more than or equal to threshold, which we set to 0.4.

We call our proposed symmetrization the bidirectional average (bidi-avg). It is easy to implement and works similarly to grow-diag-final in the sense that it tries to find an intermediate alignment between union and intersection. Figure 5 shows an example where a Japanese-to-English span prediction (left) and an English-to-Japanese span predic-

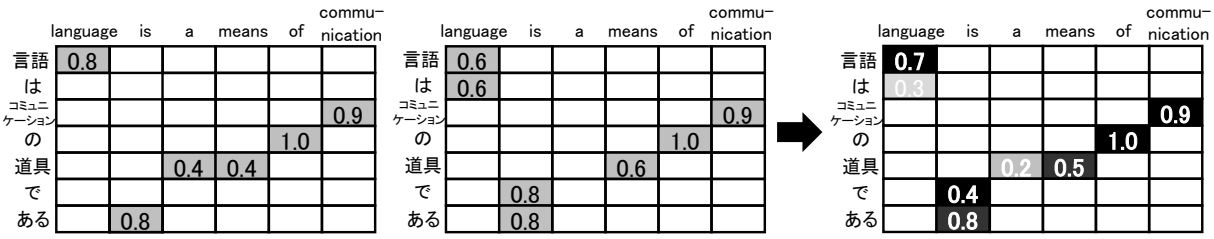


Figure 5: Ja-to-En span prediction (left) and En-to-Ja span prediction (middle) are symmetrized using bidi-avg (right). Alignments whose average probabilities are more than or equal to the threshold are shown in black.

| Language | Train | Test | Reserve |
|---|---|---|---|
| Zh-En | 4,879 | 610 | 610 |
| Ja-En | 653 | 357 | 225 |
| De-En | 300 | 208 | 0 |
| Ro-En | 150 | 98 | 0 |
| En-Fr | 300 | 147 | 0 |

Table 1: Number of gold alignment sentences and their training/test splits.

tion (middle) are symmetrized using bidirectional average (right). The token pair "is" and "で" is identified as aligned because its bidirectional average probability equals the threshold, even though it is only predicted in one direction.

We determined a threshold of 0.4 in a preliminary experiment in which we divided the Japanese-English training data into two halves for training and test sets. We used the threshold for all the experiments described in this paper. Although the span prediction of each direction was made independently, we did not normalize the scores because both directions are trained in one model.

Although we only used the best span for each direction, we could use the n-best spans to handle discontinuous alignment such as a pair between "never" and "決して... ない". It is worth investigating further as future work.

## 3 Experiments

### 3.1 Data

Table 1 shows the number of training and test sentences of the five gold word alignment datasets used in our experiments: Chinese-English (Zh-En), Japanese-English (Ja-En), German-English (De-En), Romanian-English (Ro-En), and English-French (En-Fr).

Stengel-Eskin et al. (2019) used the Zh-En dataset and Garg et al. (2019) used the De-En, Ro-En, and En-Fr datasets. We added a Ja-En dataset because Japanese is one of the most distant languages from English[3].

The Zh-En data were obtained from the GALE Chinese-English Parallel Aligned Treebank (Li et al., 2015), which consists of broadcasting news, news wires, and web data. To make the experiment's condition as close as possible to Stengel-Eskin et al. (2019), we used Chinese character-tokenized bitexts, which we cleaned (by removing mismatched bitexts, time stamps, etc.) and randomly split them into 80% training, 10% testing, and 10% future reserves.

The Japanese-English data were obtained from the KFTT word alignment data (Neubig, 2011). The Kyoto Free Translation Task (KFTT) [4] was made by manually translating Japanese Wikipedia pages about Kyoto into English. KFTT is one of the most popular Japanese-English translation benchmarks and consists of 440k training sentences, 1166 development sentences, and 1160 test sentences. The KFTT word alignment data were made by manually word aligning a part of the dev and test sets. The aligned dev set has eight files and the aligned test set has seven files. We used all eight dev set files for training, four test set files for testing, and three other files for future reserves.

De-En, Ro-En, and En-Fr data are the same ones described in Zenkel et al. (2019). They provide pre-processing and scoring scripts[5]. Garg et al. (2019) used these three datasets for their experiments. The De-En data were originally provided by Vilar et al. (2006)[6]. Ro-En and En-Fr data were used in the shared task of the HLT-NAACL-2003 workshop on Building and Using Parallel Texts (Mihalcea and Pedersen,

[3] Stengel-Eskin et al. (2019) also used an Arabic-English (Ar-En) dataset. We did not use it here due to time constraints

[4] http://www.phontron.com/kftt/index.html

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

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

2003)[7]. The En-Fr data were originally provided by (Och and Ney, 2000). The numbers of the test sentences in the De-En, Ro-En, and En-Fr datasets are 508, 248, and 447. In De-En and En-Fr, we used 300 sentences for training. In Ro-En, we used 150 sentences for training. The other sentences were used for testing.

### 3.2 Implementation Details

We used BERT-Base, Multilingual Cased (104 languages, 12 layers, 768 hidden states, 12 heads, 110M parameters, November 23rd, 2018) in our experiments[8]. We basically used the script for SQuAD as it is except for the start and end positions. The following are the parameters: train_batch_size = 12, learning_rate = 3e-5, num_train_epochs = 2, max_seq_length = 384, max_query_length = 160, and max_answer_length = 15.

Devlin et al. (2019) used the following threshold for the squad-2.0 model:

$$\hat{s_{ij}} > s_{null} + \tau \tag{4}$$

Here, if the difference of the scores of best non-null span $\hat{s_{ij}}$ and null (no-answer) span $s_{null}$ exceeds threshold $\tau$, a non-null span is predicted. The default value of $\tau = 0.0$, and the optimal threshold is decided by the development set. We used the default value because we assumed the score of a null alignment is appropriately estimated since there are many null alignments in the training data.

We used two NVIDIA TESLA V100 (16GB) for our experiments. If we set the training batch size to 6, the experiments could be performed in NVIDIA GEFORCE RTX 2080 Ti (11GB) with no significant differences in accuracy. It took about 30 minutes to fine-tune an epoch for the Ja-En data (653 sentences). It took 3 to 4 sentences per second for the inferences, excluding the time for loading the model, which was about two minutes.

### 3.3 Measures for Word Alignment Quality

We evaluated the quality of the word alignment using an F1 score that assigns equal weights to precision (P) and recall (R):

$$F_1 = 2 \times P \times R/(P + R) \tag{5}$$

If necessary, we also used alignment error rate (AER) (Och and Ney, 2003) because some previous works only reported it. Let quality of alignment $A$ be measured against a gold word alignment that contains sure ($S$) and possible($P$) alignments ($S \subseteq P$). Precision, recall, and AER are defined as follows:

$$Precision(A, P) = \frac{|P \cap A|}{|A|} \tag{6}$$

$$Recall(A, S) = \frac{|S \cap A|}{|S|} \tag{7}$$

$$AER(S, P, A) = 1 - \frac{|S \cap A| + |P \cap A|}{|S| + |A|} \tag{8}$$

Fraser and Marcu (2007) pointed out that since AER is broken in a way that favors precision, it should be used sparingly. In previous works, Stengel-Eskin et al. (2019) used precision, recall, and F1, while Garg et al. (2019) and Zenkel et al. (2019) used precision, recall, and AER. Note that, if we distinguish between sure and possible alignments, precision and recall are different from those when we do not make such a distinction. Among our five datasets, De-En and En-FR make a distinction between sure and possible alignments.

### 3.4 Results

Table 2 compares our proposed method with previous works. In all five datasets, our method outperformed all previous methods. In the Zh-En data, our method achieved an F1 score of 86.7, which is 13.3 points higher than that of DiscAlign 73.4, as reported in (Stengel-Eskin et al., 2019), which is the state-of-the-art supervised word alignment method. Stengel-Eskin et al. (2019) used 4M bitexts for pretraining, while our method needed no bitexts for pretraining. In Ja-En data, our method achieved an F1 score of 77.7, which is 20 points higher than that of GIZA++ 57.8, as reported in (Neubig, 2011).

For the De-EN, Ro-EN, and En-Fr datasets, Garg et al. (2019), which is the state-of-the-art unsupervised method, only reported AER in their paper. We classified their method as unsupervised because they did not use manually created word alignment data. Their method used the GIZA++ output for supervision. For reference, we show the precision, recall, and AER of MGIZA (Zenkel et al., 2019)[9], the AER of

[7] http://web.eecs.umich.edu/ mihalcea/wpt/index.html
[8] https://github.com/google-research/bert
[9] We took these numbers from their GitHub.

| Test set | Method | P | R | F1 | AER |
|---|---|---|---|---|---|
| Zh-En | FastAlign (Stengel-Eskin et al., 2019) | 80.5 | 50.5 | 62.0 | - |
| | DiscAlign (Stengel-Eskin et al., 2019) | 72.9 | 74.0 | 73.4 | - |
| | Our method | 84.4 | 89.2 | **86.7** | - |
| Ja-En | Giza++ (Neubig, 2011) | 59.5 | 55.6 | 57.6 | - |
| | Our method | 77.3 | 78.0 | **77.6** | - |
| De-En | Our method (trained on sure + possible) | 89.9 | 81.7 | 85.6 | - |
| Ro-En | Our method | 90.4 | 85.3 | 87.8 | - |
| En-Fr | Our method (only trained on sure) | 79.6 | 93.9 | 86.2 | - |
| De-En | MGIZA (BPE, Grow-Diag-Final) (Zenkel et al., 2019) | 91.3 | 70.2 | - | 20.6 |
| | GIZA++ (BPE, Grow-Diag) (Zenkel et al., 2020) | - | - | - | 18.7 |
| | Alignment layer, bidi, unsupervised (Zenkel et al., 2020) | - | - | - | 16.3 |
| | Align and translate, GIZA++ supervised (Garg et al., 2019) | - | - | - | 16.0 |
| | Our method (trained on sure + possible) | 89.9 | 87.3 | - | **11.4** |
| Ro-En | MGIZA (BPE, Grow-Diag-Final) (Zenkel et al., 2019) | 90.9 | 61.8 | - | 26.4 |
| | GIZA++ (BPE, Grow-Diag) (Zenkel et al., 2020) | - | - | - | 26.5 |
| | Alignment layer, bidi, unsupervised (Zenkel et al., 2020) | - | - | - | 23.4 |
| | Align and translate, GIZA++ supervised (Garg et al., 2019) | - | - | - | 23.1 |
| | Our method | 90.4 | 85.3 | - | **12.2** |
| En-Fr | MGIZA (BPE, Grow-Diag) (Zenkel et al., 2019) | 97.5 | 89.7 | - | 5.9 |
| | GIZA++ (BPE, Grow-Diag) (Zenkel et al., 2020) | - | - | - | 5.5 |
| | Discriminative matching (Taskar et al., 2005) | - | - | - | 5.4 |
| | Supervised ITG (Haghighi et al., 2009) | 95.5 | 94.2 | - | 5.0 |
| | Alignment layer, bidi, unsupervised (Zenkel et al., 2020) | - | - | - | 5.0 |
| | Align and translate, GIZA++ supervised (Garg et al., 2019) | - | - | - | 4.6 |
| | Our method (only trained on sure) | 97.7 | 93.9 | - | **4.0** |

Table 2: Best-effort comparison of proposed method with previous works

GIZA++ (Zenkel et al., 2020), as well as the accuracies of previous methods (Taskar et al., 2005; Haghighi et al., 2009; Zenkel et al., 2020) with the same datasets.

For training, we used both the sure and possible alignments for the De-En dataset, but we only used sure alignments for the En-Fr dataset because it is very noisy[10].

We used the scoring script provided by Zenkel et al. (2019). For the De-En, Ro-En, and En-Fr datasets, the AERs of the proposed method were 11.4, 12.2, and 4.0, which are significantly smaller than those of (Garg et al., 2019) and (Zenkel et al., 2020).

It is unfair to compare our supervised method with unsupervised methods. Our experiment's aim is to show that we have made supervised methods practical. We can train our model using a smaller amount of manually created data than the amount originally created for evaluation.

## 4 Analysis

### 4.1 Symmetrization Heuristics

To show the effectiveness of our proposed symmetrization heuristics (bidi-avg), Table 3 describes the word alignment accuracies of the predictions of two directions, intersection, unison, grow-diag-final, and bidi-avg.

The accuracies are greatly affected by the orthography of the target language. For languages whose words are not delimited by white spaces, such as Chinese and Japanese, the span prediction accuracy "to English" is significantly higher than that of "from English". In this case, grow-diag-final outperforms bidi-avg. By contract, for languages with spaces between words, such as Ger-

[10]In En-Fr data, all "phrasal correspondence" are annotated as possible alignments. For example, if a phrase with three words and a phrase with four words are regarded as mutual translations, 12 word alignments are marked as possible (Mihalcea and Pedersen, 2003). There are 4,038 sure alignments and 13,400 possible alignments in the En-Fr data (447 sentences). If our model is trained on both sure and possible, such numerous possible alignments function as noise, which results in low precision.

| Test set | Method | P | R | F1 |
|---|---|---|---|---|
| Zh-En | Zh to En | 89.9 | 85.8 | **87.8** |
| | En to Zh | 82.0 | 81.8 | 81.9 |
| | intersection | 95.5 | 74.9 | 83.9 |
| | union | 79.4 | 92.7 | 85.5 |
| | grow-diag-final | 94.7 | 81.2 | 87.4 |
| | bidi-avg | 84.4 | 89.2 | 86.7 |
| Ja-En | Ja to En | 80.6 | 79.7 | **80.2** |
| | En to Ja | 61.9 | 69.0 | 65.2 |
| | intersection | 90.8 | 63.1 | 74.5 |
| | union | 60.8 | 85.6 | 71.1 |
| | grow-diag-final | 86.5 | 71.7 | 78.1 |
| | bidi-avg | 77.3 | 78.0 | 77.6 |
| De-En | De to En | 86.7 | 80.7 | 83.6 |
| | En to De | 87.0 | 82.1 | 84.5 |
| | intersection | 93.8 | 76.1 | 84.0 |
| | union | 81.5 | 86.7 | 84.0 |
| | grow-diag-final | 91.1 | 78.4 | 84.3 |
| | bidi-avg | 81.7 | 89.9 | **85.6** |
| Ro-En | Ro to En | 84.6 | 86.5 | 85.5 |
| | En to Ro | 87.2 | 86.3 | 86.7 |
| | intersection | 93.1 | 82.2 | 87.3 |
| | union | 80.2 | 90.6 | 85.0 |
| | grow-diag-final | 92.0 | 83.7 | 87.6 |
| | bidi-avg | 90.4 | 85.3 | **87.8** |
| En-Fr | En to Fr | 79.9 | 91.7 | 85.4 |
| | Fr to En | 79.5 | 91.3 | 85.0 |
| | intersection | 85.3 | 88.1 | **86.7** |
| | union | 75.2 | 94.9 | 83.9 |
| | grow-diag-final | 79.6 | 92.4 | 85.5 |
| | bidi-avg | 79.6 | 93.9 | 86.2 |

Table 3: Effects of symmetrization for various language pairs

man, Romanian, and French, no significant differences exist between the "to English" and "from English" accuracies. In this case, bidi-avg is better than grow-diag-final. In En-Fr, intersection achieves the best accuracy, probably because the dataset is very noisy. Since the proposed bidi-avg works relatively well for all cases, we used the heuristics as the default symmetrization method in our experiments.

### 4.2 Importance of Source Context

Table 4 shows the word alignment accuracies for questions of different source contexts. We used the Ja-En data and found that the source context information is critical for predicting the target span. Without it, the F1 score of the proposed method

| Test set | Context | P | R | F1 |
|---|---|---|---|---|
| Ja-En | no context | 67.3 | 53.0 | 59.3 |
| | $\pm$2 words | 73.9 | 70.2 | 72.0 |
| | whole sentence | 77.3 | 78.0 | 77.6 |

Table 4: Importance of source context

| Test set | # train | P | R | F1 |
|---|---|---|---|---|
| Zh-En | 300 | 80.9 | 78.4 | 79.6 |
| | 600 | 82.9 | 81.7 | 82.3 |
| | 1200 | 82.8 | 85.6 | 84.1 |
| | 2400 | 83.6 | 87.4 | 85.5 |
| | 4879 | 84.4 | 89.2 | 86.7 |

Table 5: Test set performance when trained on subsamples of Chinese gold word alignment data

is 59.3, which is only slightly higher than that of GIZA++, 57.6. If we add a short context, namely, the two preceding words and the two following words, the F1 score is improved by more than 10 points to 72.0. If we use the whole source sentence as the context, the F1 score is improved by 5.6 points to 77.6.

It is nontrivial to obtain accurate alignments from a set of independently predicted spans. In preliminary experiments, we used Integer Linear Programming (ILP) to optimize the span predictions as in (DeNero and Klein, 2008). We found that using context is simple and more effective.

### 4.3 Learning Curve

Table5 shows the learning curve of the proposed method using the Zh-En data. Compared to previous methods, our method achieved higher accuracy using less training data. Even for 300 sentences, the F1 score of our method was 79.6, which is 6.2 points higher than that of (Stengel-Eskin et al., 2019) (73.4), which used more than 4800 sentences for training.

A supervised model trained on hand-aligned data must learn the idiosyncrasies of the annotation standard, which varies widely from language to language and across different annotation efforts. Our method allows us to fine-tune the specific peculiarities of the annotation standard using only a few hundred examples.

## 5 Related Works

For several years, word alignment methods using neural networks (Yang et al., 2013; Tamura et al., 2014; Legrand et al., 2016) failed to signifi-

cantly outperform those using statistical methods (Brown et al., 1993; Vogel et al., 1996).

Recently, Stengel-Eskin et al. (2019) proposed a supervised method using a small amount of annotated data (1.7K-5K sentences) and significantly outperformed the accuracy of GIZA++. They first mapped the source and target word representations obtained from the encoder and decoder of the Transformer to a shared space using a three-layer feed-forward neural network. They then applied 3 $\times$ 3 convolution and softmax to obtain the alignment scores between the source and target words. They used 4M parallel sentences to pretrain the Transformer. We achieved significantly better accuracy than (Stengel-Eskin et al., 2019) with less annotated training data and no parallel sentences.

Garg et al. (2019) proposed an unsupervised method that jointly optimized translation and alignment objectives. They achieved a significantly better alignment error rate (AER) than GIZA++ when they supervised their model using the alignments obtained from GIZA++. Their model requires about a million parallel sentences for training the underlying Transformer. Zenkel et al. (2020) added an alignment layer on top of the Transformer, which uses full target context and a loss function to encourage contiguous alignment and bidirectional agreement. Their unsupervised end-to-end neural word alignment method consistently outperformed GIZA++. We experimentally showed that we can outperform previous unsupervised neural word alignment results with just 150 to 300 annotated sentences for training. Although it is not fair to compare unsupervised methods with supervised ones, our method is a practical option to obtain better word alignment results.

Ouyang and McKeown (2019) proposed a monolingual phrase alignment method that can align phrases of arbitrary lengths. Compared to our span prediction method, their method is inflexible because they first segmented the source and target sentences into chunks and used a pointer-network (Vinyals et al., 2015) to calculate the alignment scores between fixed chunks.

Cao et al. (2020) reported that multilingual BERT is somewhat aligned out-of-the-box, and proposed a method to align pretrained contextual word embeddings. Their method learns a function that maps aligned word pairs to similar representations, while our method implicitly learns a function that maps a word representation to its translation with both contexts.

## 6 Conclusion

We presented a novel supervised word alignment method using the multilingual BERT, which requires as few as 300 training sentences to outperform previous supervised and unsupervised methods. We made supervised word alignment practical because our method does not require any bitexts for pretraining, and it can be fine-tuned to a specific guideline using fewer gold word alignments.

Future works include using other multilingual pretraining models such as XLM-RoBERTa (Conneau et al., 2019) for a more accurate model and distilmBERT (Sanh et al., 2019) for a more compact model. One significant merit of framing word alignment as a SQuAD-style span prediction task is that we can easily import the progress of the latest question answering and multilingual language modeling technologies.

Our cross-language span prediction method can be used for any alignments between two sequences. We have already applied it to bilingual sentence alignment (Chousa et al., 2020) and we plan to extend it to other related problems.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the ICLR-2015*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual Alignment of Contextual Word Representations. In *Proceedings of the ICLR-2020*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the EMNLP-2014*, pages 1724–1734.

Katsuki Chousa, Masaaki Nagata, and Masaaki Nishino. 2020. SpanAlign: Sentence Alignment Method based on Cross-Language Span Prediction and ILP. In *Proceedings of the COLING-2020 (to appear)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. ArXiv:1911.02116.

John DeNero and Dan Klein. 2008. The Complexity of Phrase Alignment Problems. In *Proceedings of the ACL-2008*, pages 25–28.

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 NAACL-2019*, pages 4171–4186.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the NAACL-HLT-2013*, pages 644–648.

Alexander Fraser and Daniel Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3):293–303.

Qin Gao and Stephan Vogel. 2008. Parallel Implementations of Word Alignment Tool. In *Proceedings of ACL 2008 workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly Learning to Align and Translate with Transformer Models. In *Proceedings of the EMNLP-IJCNLP-2019*, pages 4452–4461.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better Word Alignments with Supervised ITG Models. In *Proceedings of the ACL-2009*, pages 923–931.

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 WMT-2019*, pages 116–127.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL-2007*, pages 177–180.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural Network-based Word Alignment through Score Aggregation. In *Proceedings of the WMT-2016*, pages 66–73.

Xuansong Li, Stephen Grimes, Stephanie Strassel, Xiaoyi Ma, Nianwen Xue, Mitch Marcus, and Ann Taylor. 2015. GALE Chinese-English Parallel Aligned Treebank – Training. Web Download. LDC2015T06.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the EMNLP-2015*, pages 1412–1421.

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.

Graham Neubig. 2011. Kyoto Free Translation Task alignment data package. http://www.phontron.com/kftt/.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL-2000*, pages 440–447.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Jessica Ouyang and Kathy McKeown. 2019. Neural Network Alignment for Sentential Paraphrases. In *Proceedings of the ACL-2019*, pages 4724–4735.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the ACL-2018*, pages 784—789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP-2016*, pages 2383–2392.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv:1910.01108.

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 NAACL-2019*, pages 449–459.

Elias Stengel-Eskin, Tzu ray Su, Matt Post, and Benjamin Van Durme. 2019. A Discriminative Neural Model for Cross-Lingual Word Alignment. In *Proceedings of the EMNLP-IJCNLP-2019*, pages 910–920.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the NIPS-2014*, pages 3104–3112.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent Neural Networks for Word Alignment Model. In *Proceedings of the ACL-2014*, pages 1470–1480.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A Discriminative Matching Approach to Word Alignment. In *Proceedings of the HLT-EMNLP-2005*, pages 73–80.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of ACL-2016*, pages 76–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of the NIPS 2017*, pages 5998–6008.

David Vilar, Maja Popović, and Hermann Ney. 2006. AER: Do we need to "improve" our alignments? In *Proceedings of IWSLT-2006*, pages 2005–212.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Proceedings of NeurIPS-2015*.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of COLING-1996*.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word Alignment Modeling with Context Dependent Deep Neural Network. In *Proceedings of the ACL-2013*, pages 166–175.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. In *Proceedings of HLT-2001*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. ArXiv:1901.11359.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-End Neural Word Alignment Outperforms GIZA++. In *Proceeding of the ACL-2020*, pages 1605–1607.