*Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 6861–6867
Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

# SAPPHIRE: Simple Aligner for Phrasal Paraphrase with Hierarchical Representation

**Masato Yoshinaka[†], Tomoyuki Kajiwara[‡], Yuki Arase[†]**
[†]Graduate School of Information Science and Technology, Osaka University
[‡]Institute for Datability Science, Osaka University
{yoshinaka.masato, arase}@ist.osaka-u.ac.jp, kajiwara@ids.osaka-u.ac.jp

**Abstract**

We present SAPPHIRE, a Simple Aligner for Phrasal Paraphrase with HIerarchical REpresentation. Monolingual phrase alignment is a fundamental problem in natural language understanding and also a crucial technique in various applications such as natural language inference and semantic textual similarity assessment. Previous methods for monolingual phrase alignment are language-resource intensive; they require large-scale synonym/paraphrase lexica and high-quality parsers. Different from them, SAPPHIRE depends only on a monolingual corpus to train word embeddings. Therefore, it is easily transferable to specific domains and different languages. Specifically, SAPPHIRE first obtains word alignments using pre-trained word embeddings and then expands them to phrase alignments by bilingual phrase extraction methods. To estimate the likelihood of phrase alignments, SAPPHIRE uses phrase embeddings that are hierarchically composed of word embeddings. Finally, SAPPHIRE searches for a set of consistent phrase alignments on a lattice of phrase alignment candidates. It achieves search-efficiency by constraining the lattice so that all the paths go through a phrase alignment pair with the highest alignment score. Experimental results using the standard dataset for phrase alignment evaluation show that SAPPHIRE outperforms the previous method and establishes the state-of-the-art performance.

**Keywords:** phrase alignment, phrasal paraphrase

## 1. Introduction

Monolingual phrase alignment is one of the fundamental tasks in natural language understanding. It identifies the most plausible phrase alignments that are semantically equivalent in a monolingual sentence pair. The applications of monolingual phrase alignment are diverse. The most relevant application is the sentence pair modeling tasks (Lan and Xu, 2018), such as recognizing textual entailment (Dagan et al., 2006) and assessing semantic textual similarity (Sultan et al., 2014). Besides, monolingual phrase alignment is useful to identify parallel sentences automatically from a comparable corpus (Kajiwara and Komachi, 2016) and to generate paraphrases (Li et al., 2019).

Previous studies on monolingual phrase alignment (MacCartney et al., 2008; Yao et al., 2013; Arase and Tsujii, 2017; Ouyang and McKeown, 2019) depend on large-scale paraphrase dictionaries or assume the availability of high-quality parsers or chunkers, which severely restricts the applicability of alignment methods to corpora of specific domains and different languages. On the other hand, *bilingual* phrase extraction that has been widely studied in the field of statistical machine translation (SMT) only assumes the availability of a large-scale parallel corpus. The standard approach is first identifying word alignment and then composing phrase alignments from the identified word alignments based on heuristics. However, the purpose of bilingual phrase extraction is the collection of a large scale bilingual phrase pairs, and thus, identification of plausible phrase alignments in a single sentence pair is out of their scope.

In this study, we propose a simple aligner for phrasal paraphrase with hierarchical representation (SAPPHIRE) that takes advantage of the bilingual phrase pair extraction approach. Specifically, SAPPHIRE identifies word alignments from pre-trained word embeddings and then composes candidates of phrase alignments based on methods developed for bilingual phrase extraction. Finally, SAPPHIRE searches a set of *consistent* phrase alignments in a sentence pair.

The contributions of this study are twofold:

- We developed a simple monolingual phrase aligner, SAPPHIRE, that depends only on a monolingual raw corpus to train word embeddings. Such a raw corpus is abundantly available for a variety of domains and languages. Therefore, SAPPHIRE is easily transferable to any domains and languages.
- SAPPHIRE can handle arbitrary units for phrases and, by default, aligns phrases of word $n$-grams. If syntactic parsers or chunkers are available, SAPPHIRE can identify alignments of phrases that conform to the desired phrase unit.

In the experiment using the English monolingual phrase alignment benchmark (Brockett, 2007), SAPPHIRE outperformed the state-of-the-art method (Ouyang and McKeown, 2019) and achieved the best phrase alignment performance. SAPPHIRE is publicly available on our web site.[1]

## 2. Phrase Alignment by SAPPHIRE

This section first defines the phrase alignment problem and then describes details of the alignment process of SAPPHIRE.

### 2.1. Problem Definition

We assume that SAPPHIRE takes a pair of input sentences $X = x_0, \ldots, x_{|X|}$ and $Y = y_0, \ldots, y_{|Y|}$, which consist of

[1] https://github.com/m-yoshinaka/sapphire

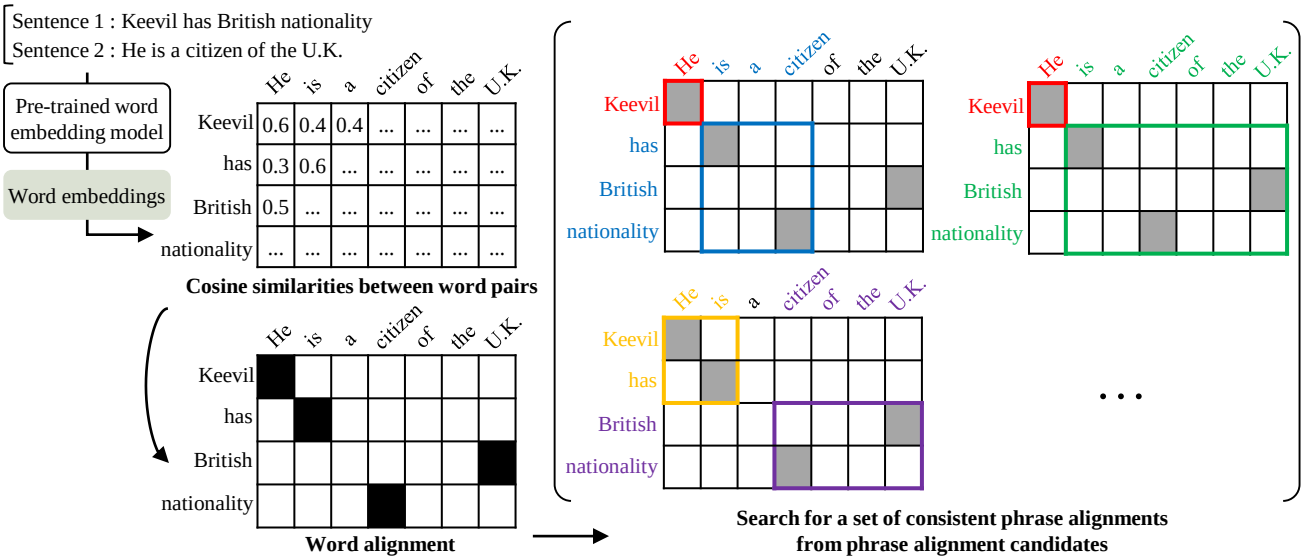


Figure 1: Overview of phrase alignment process by SAPPHIRE

$|X|$ and $|Y|$ words, respectively. SAPPHIRE identifies a set of consistent phrase alignments $A = \{a_k = (x_p^q, y_r^s) | x_p^q \in X, y_r^s \in Y\}$, where $x_p^q = x_p, \ldots, x_q$ is a phrase in $X$ starting at the $p$-th word and spanning to the $q$-th word. Similarly, $y_r^s = y_r, \ldots, y_s$ is a phrase in $Y$. We allow a phrase not to have an alignment (*i.e.*, null alignment), and hence $x_p^q$ or $y_r^s$ can be *null* ($\emptyset$).

We define a set of consistent phrase alignments as one-to-one phrase alignments that do not overlap each other. More formally, a set of consistent alignments $A$ must satisfy that for $\exists a_k = (x_p^q, y_r^s), \exists a_l = (x_f^g, y_m^n) \in A\ (k \neq l)$, the span of $p$ to $q$ does not overlap with the span of $f$ to $g$ in $X$, and similarly, the span of $r$ to $s$ does not overlap with the span of $m$ to $n$ in $Y$.[2]

A sentence pair can have multiple sets of consistent alignments. The number of possible combinations of phrase alignments exponentially grows as the sentences become longer. Hence, it is computationally intractable to determine the most plausible set of consistent alignments. Therefore, SAPPHIRE finds an approximate solution using a search method designed based on the characteristics of phrasal paraphrases.

## 2.2. Overview of Alignment Process

Figure 1 shows the overview of the phrase alignment process of SAPPHIRE. First, SAPPHIRE obtains word alignments based on cosine similarities between every pair of word embeddings in the sentence pair. Next, it extends the word alignments to phrase alignments using methods for bilingual phrase extraction. Finally, SAPPHIRE searches a set of consistent phrase alignments on a lattice constructed from phrase alignment candidates.

## 2.3. Embedding-based Word Alignment

SAPPHIRE obtains word alignment candidates based on the grow-diag-final heuristic (Koehn et al., 2003) designed for bilingual phrase alignment and the extended Hungarian algorithm for rectangle matrices (Bourgeois and Lassalle, 1971) using cosine similarities between word embeddings. Word alignments obtained by the grow-diag-final heuristic and the Hungarian algorithm might be unreliable. Hence, SAPPHIRE selects final word alignments among the obtained alignment candidates whose cosine similarities are greater than or equal to a threshold $\lambda$, like Song and Roth (2015).

**Grow-Diag-Final Alignment** In the method using the grow-diag-final heuristic, SAPPHIRE first computes alignments based on cosine similarities of word embeddings from both $X$ to $Y$ and $Y$ to $X$ directions. Then it obtains the final candidates of word alignments following the grow-diag-final heuristic. SAPPHIRE associates $x_i$ to $y_j$ that has the highest cosine similarity:

$$(x_i, y_j) = \arg\max_k \cos(\mathbf{e}_{x_i}, \mathbf{e}_{y_k}), \quad (1)$$

where $\mathbf{e}_{x_i}$ and $\mathbf{e}_{y_j}$ are word embeddings of $x_i$ and $y_j$, respectively. Similarly, it associates $y_j$ to $x_l$:

$$(y_j, x_l) = \arg\max_k \cos(\mathbf{e}_{y_j}, \mathbf{e}_{x_k}). \quad (2)$$

Following the grow-diag-final heuristic, an initial set of word alignment candidates are the intersection of alignments from both directions. Next, the grow-diag-final heuristic adds alignments from the union set, considering its association matrix if they meet the following conditions.

- Alignments whose words are adjacent in the vertical, horizontal, or diagonal directions of alignments in the initial candidate set.
- Alignments whose words have no alignment in the initial candidate set.

Note that the final set of word alignment candidates have many-to-many alignments.

[2] We regard a phrase of null does not have a span, and hence it does not overlap with any phrases.

**Algorithm 1** Extraction of Phrase Alignment Candidates

**Input:** Index pairs of word alignments $W = \{(i, j)\}$, a sentence pair of $X$ and $Y$
**Output:** Phrase alignment candidates $U = \{(x_p^q, y_r^s)\}$

1: Initialization: $U \leftarrow \emptyset$, $M \leftarrow \mathbb{0}$
2: **for** $(i, j)$ in $W$ **do**
3: $\quad M_{i,j} \leftarrow 1$ ▷ Create a word alignment matrix
4: **for** $(i, j), (i', j')$ in $W$ **do**
5: $\quad i_s = \min(i, i'), \quad i_e = \max(i, i')$
6: $\quad j_s = \min(j, j'), \quad j_e = \max(j, j')$
7: $\quad u \leftarrow (x_{i_s}^{i_e}, y_{j_s}^{j_e})$
8: $\quad$ **while do**
9: $\quad\quad$ **if** $u$ has adjacent word alignments $M_s$ in vertical and horizontal directions in $M$ **then**
10: $\quad\quad\quad$ **for** $(k, l) \in M_s$ **do**
11: $\quad\quad\quad\quad i_s = \min(i_s, k), \quad i_e = \max(i_e, k)$
12: $\quad\quad\quad\quad j_s = \min(j_s, l), \quad j_e = \max(j_e, l)$
13: $\quad\quad\quad\quad u \leftarrow (x_{i_s}^{i_e}, y_{j_s}^{j_e})$
14: $\quad\quad$ **else break**
15: $\quad U \leftarrow U \cup \{u\}$

**Hungarian Alignment** The Hungarian algorithm is the optimization algorithm that solves the cost assignment problem. SAPPHIRE sets the cost matrix $C$ as cosine similarities of all word pairs and then obtains the one-to-one word alignments by the Hungarian algorithm.
The cost of each word pair $(x_i, y_j)$ is

$$\text{cost}(x_i, y_j) = 1 - \cos(\mathbf{e}_{x_i}, \mathbf{e}_{y_j}). \tag{3}$$

The Hungarian algorithm minimizes the cost in the cost matrix as

$$\min \sum_i \sum_j C_{i,j} Z_{i,j}, \tag{4}$$

where $C_{i,j}$ is the cost of between $x_i$ and $y_j$, and $Z$ is the final word alignment matrix. $Z_{i,j} = 1$ if row $i$ and column $j$ is assigned, *i.e.*, $x_i$ is aligned to $y_j$, and $Z_{i,j} = 0$, otherwise.

## 2.4. Extraction of Phrase Alignment Candidates

SAPPHIRE obtains phrase alignment candidates by expanding the word alignments based on the bilingual phrase alignment heuristic used by Moses (Koehn et al., 2007). Algorithm 1 presents the algorithm of phrase alignment candidate extraction. It generates a phrase alignment that covers an arbitrary pair of word alignments (lines 5 to 7). It expands the phrase alignment if there are adjacent word alignments (lines 8 to 14). As Algorithm 1 shows, a phrase means a word $n$-gram, which can be a single word to the entire sentence. However, we can easily adapt to align grammatical phrases by restricting phrase alignment candidates to conform to predetermined spans of phrases.
For each pair of phrase alignment candidates, SAPPHIRE computes a score to estimate the likelihood of the alignment. In bilingual phrase alignment, such scores are translation probabilities. SAPPHIRE computes the score based on phrase embedding composed of word embeddings hierarchically. After calculating scores of all phrase pairs, SAPPHIRE filters out unreliable alignments whose scores are less than a threshold $\delta$.

In this study, we use simple mean-pooling of word embeddings to generate a phrase embedding. SAPPHIRE computes the score of alignment likelihood based on cosine similarity between phrase embeddings. Because of the simple pooling method, the pure cosine similarity becomes small when a phrase pair is longer. To complement this, SAPPHIRE biases the cosine similarity to consider the length of phrases. Accurately, SAPPHIRE computes the score to align phrase $x$ and $y$ as

$$\text{score}(x, y) = \cos(\mathbf{e}_x, \mathbf{e}_y) - \alpha \cdot \frac{1}{|x| + |y|}, \tag{5}$$

where $\mathbf{e}_x$ and $\mathbf{e}_y$ are phrase embeddings generated by mean-pooling of word embeddings contained in $x$ and $y$, respectively. The function $|\cdot|$ computes the length of a phrase, and $\alpha$ is a hyperparameter controlling the weight of bias toward phrase lengths.
For example, we have two unigram phrase pairs of *New* $\leftrightarrow$ *New* and *York* $\leftrightarrow$ *York*, as well as a bigram phrase pair of *New York* $\leftrightarrow$ *New York*. When $\alpha$ is large, the bigram phrase pair of *New York* $\leftrightarrow$ *New York* receives a higher alignment score than the average of alignment scores of the *New* $\leftrightarrow$ *New* and *York* $\leftrightarrow$ *York*.

## 2.5. Searching the Consistent Phrase Alignments

Finally, SAPPHIRE identifies a set of consistent phrase alignments from the obtained phrase alignment candidates. As discussed in Section 2.1, it is computationally intractable to enumerate all the possible combinations of phrase alignment candidates. For computational efficiency, SAPPHIRE constructs a lattice that satisfies the definitions of consistent phrase alignments and searches for the approximate solution to obtain a plausible set of alignments.
Our observation of phrase correspondences in sentence pairs found that difficulty of phrase alignments are not uniform; there are easier and harder alignments to identify. Based on this observation, we designed a search algorithm that prioritizes the phrase alignment of the highest score, as Figure 2 shows.
SAPPHIRE first identifies the phrase pair with the highest score as the starting node to construct a lattice. It then adds alignment candidates that do not overlap each other in both forward and backward directions into the lattice, as shown in Figure 2. While constructing the lattice, SAPPHIRE dynamically traverses all the paths by depth-first search and outputs the one with the highest average alignment score. Because the lattice is constrained so that all the paths go through the first phrase pair, the computational costs for traversing are small. Our preliminary experiment confirmed that this approach was superior to a commonly used left-to-right searching approach.

# 3. Experiment

We evaluate the performance of SAPPHIRE using the standard dataset for phrase alignment evaluation through a comparison to the current state-of-the-art method (Ouyang and McKeown, 2019).

## 3.1. Dataset

We use the Microsoft Research Recognizing Textual Entailment (MSR RTE) corpus (Brockett, 2007) as the standard

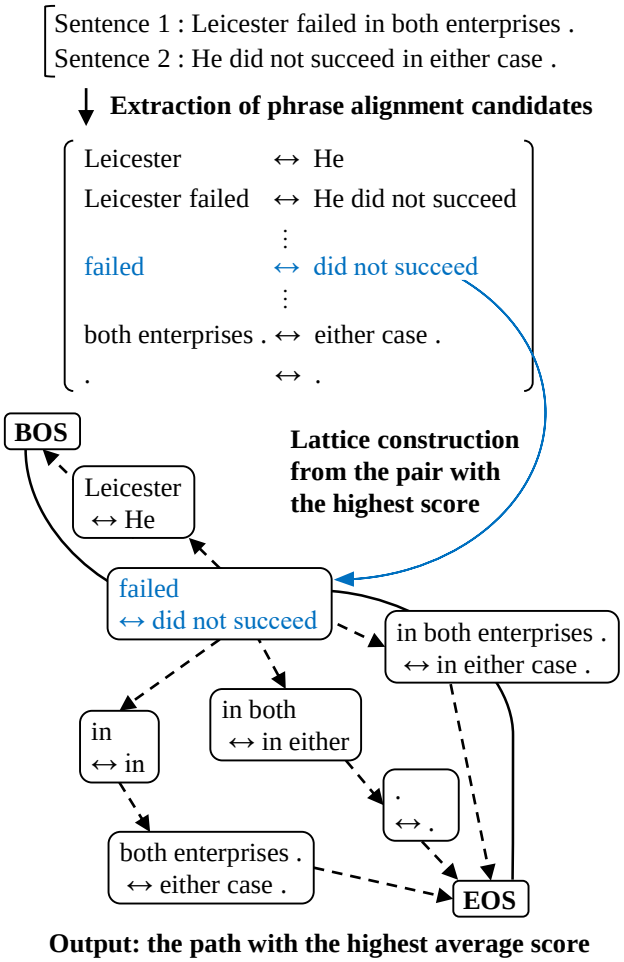


Figure 2: Lattice construction from extracted phrase alignment candidates and searching for the plausible phrase alignment. Solid edges represent the path to be output.

evaluation corpus of phrase alignment. MSR RTE corpus manually annotates 2006 PASCAL RTE2 corpus (Dagan et al., 2006; Bar-Haim et al., 2006), and consists of a development set and a test set, each of which contains 800 sentence pairs. Three annotators assigned many-to-many word alignments, which are convertible to phrase alignments. The alignments have two types: sure and possible. Sure alignments are alignments that the annotators were confident, such as pairs of the same words and synonyms. The possible alignments are alignments that the annotators were not as confident as sure alignments, but likely to have alignments.

Previous studies of monolingual phrase alignment evaluated using only sure alignments, but most of the sure alignments are one-to-one. Hence, Yao et al. (2013) created the phrase alignments by merging consecutive sure alignments, but the percentage of alignments with phrases of more than 4 words was only 1%.

We follow Ouyang and McKeown (2019) and evaluate on annotations with an increased phrase alignment ratio by utilizing possible alignments. In particular, we decide sure and possible alignment by a majority voting of annotations by the three annotators. We use only the sentence pairs which contain at least one possible alignment. For each 800 sentence pairs in the development and test sets, 487 and 441 sentence pairs contain at least one possible alignment and used for evaluation, respectively.

| Symbol | Role of hyperparameter |
|---|---|
| $\lambda$ | Prunes word alignment candidates |
| $\delta$ | Prunes phrase alignment candidates |
| $\alpha$ | Biases the phrase alignment score based on the lengths of phrases |

Table 1: Summary of hyperparameters

### 3.2. Evaluation Metrics

Previous studies evaluate the quality of the phrase alignments by measuring the quality of word alignments inside (Yao et al., 2013; Ouyang and McKeown, 2019). We also use the precision, recall, and F-measure of word alignments as evaluation metrics following the previous studies. For our phrase alignment results, we regard that every word pair in a phrase alignment has an alignment, following Ouyang and McKeown (2019).

### 3.3. Implementation Details

As word embedding models, we investigate the effects of static and dynamic embeddings. As the static word embeddings, we use the pre-trained model of fastText (Bojanowski et al., 2017).[3] As the dynamic word embeddings, we use the pre-trained model of BERT (Devlin et al., 2019) that generates embeddings from the contexts.[4] Specifically, we use the output of the last layer of each token.

Table 1 summarizes the hyperparameters in SAPPHIRE. We tuned these hyperparameters by a grid search to maximize the F-measure score at the development set. We searched settings of $\lambda$ and $\delta$ from the range of [0.5, 0.9] with 0.1 intervals, and $\alpha$ from [0.05, 0.10] with 0.01 intervals.

### 3.4. Results

Table 2 shows the results of the phrase alignment evaluation on the test set of MSR RTE corpus.

The scores of the method proposed by Ouyang and McKeown (2019) are borrowed from their paper, which is the current state-of-the-art. Overall, SAPPHIRE using fastText as the word embedding model and the Hungarian algorithm for word alignment performed the best, which outperformed (Ouyang and McKeown, 2019) by 4.3% on F-measure. You may think that the precision, recall, and F-measure scores are low even on the best results. This is because we artificially aligned every word pair in a phrase alignment (Section 3.2), which does not necessarily happen in practice.

When we compare the performances of fastText and BERT in SAPPHIRE, fastText shows a much higher F-measure

[3] wiki-news-300d-1M-subword: https://fasttext.cc/docs/en/english-vectors

[4] bert-base-uncased: https://github.com/google-research/bert

| Method | Word Embedding | Word Alignment | P% | R% | $F_1$% |
|---|---|---|---|---|---|
| (Ouyang and McKeown, 2019) | – | – | 23.4 | **47.7** | 31.4 |
| SAPPHIRE | fastText | grow-diag-final | 31.6 | 40.6 | 35.5 |
| | fastText | Hungarian | **32.0** | 40.2 | **35.7** |
| | BERT | grow-diag-final | 12.9 | 35.4 | 18.9 |
| | BERT | Hungarian | 13.0 | 35.0 | 18.9 |

Table 2: Evaluation results on the MSR RTE test set

All that changed in 1922 , when **Tutankhamun 's tomb was discovered by** *Egyptologist* **Howard Carter** on behalf of his patron **Lord Carnarvon .**

**Tutankhamun 's Tomb was unearthed by *Howard Carter*** and **Lord Carnarvon .**

(a) Gold phrase alignment (sure alignments in **bold** and possible alignments in *italic*)

All that changed in 1922 , when Tutankhamun 's tomb was discovered by Egyptologist Howard Carter on behalf of his patron Lord Carnarvon .

Tutankhamun 's Tomb was unearthed by Howard Carter and Lord Carnarvon .

(b) Phrase alignment output by SAPPHIRE

The ROE printed here were issued by General Jean Cot , then commander of **U.N. *forces*** , and were intended to establish the conditions under which the forces could use their weapons as they carry out the U.N. peacekeeping mission in **Bosnia** .

**U.N.** *peacekeeping* **forces** withdrew from **Bosnia** .

(c) Gold phrase alignments on a sentence pair with a large length difference (sure alignments in **bold** and possible alignments in *italic*)

The ROE printed here were issued by General Jean Cot , then commander of U.N. forces , and were intended to establish the conditions under which the forces could use their weapons as they carry out the U.N. peacekeeping mission in Bosnia .

U.N. peacekeeping forces withdrew from Bosnia .

(d) Phrase alignment output by SAPPHIRE on a sentence pair with a large length difference

Figure 3: Examples of phrase alignments on MSR RTE corpus

score than BERT. This result is from the side-effect of contextualized word embeddings; words in semantically similar sentences tend to have closer embeddings (Ethayarajh, 2019). Because most sentence pairs in MSR RTE corpus are semantically relevant due to its purpose of RTE, the side-effect of contextualized word embeddings should be pronounced.

When we compare word alignment methods of the grow-diag-final heuristic and the Hungarian algorithm, the former has a slightly higher recall but lower precision, and the latter has higher precision but lower recall. These characteristics are more noticeable when $\lambda$ is small.

Figure 3 shows a couple of examples of the phrase alignment by SAPPHIRE with fastText embedding. Figure 3 (a) and 3 (b) show that SAPPHIRE correctly identifies most of the gold phrase alignments except the alignment of *tomb* $\leftrightarrow$ *Tomb*. This error is because the word *Tomb* was an unknown word due to its capitalization. It can be easy to make the word alignment process more robust by including fuzzy matching of words based on their surface similarities. The sentence pairs in Figure 3 (c) and 3 (d) have a large difference in their sentence lengths, which causes most of the phrases in the first sentence should be unaligned. Besides, the first sentence has two similar phrases of *U.N. forces* and *U.N. peacekeeping*to the phrase of *U.N. peacekeeping forces* in the second sentence. For such a challenging sentence pair, SAPPHIRE correctly identifies the alignment *U.N. forces* $\leftrightarrow$ *U.N. peacekeeping forces*.

## 4. Related Work

There have been two approaches in monolingual phrase alignment; one aligns arbitrary phrases without grammat-

ical constraints, and the other aligns phrases defined by a grammar. In any case, previous methods of monolingual phrase alignment are resource-intensive. As the first approach, MANLI (MacCartney et al., 2008) and the following studies (Thadani and McKeown, 2011; Thadani et al., 2012), as well as Yao et al. (2013), depend on lexical database of WordNet (Miller, 1995) or paraphrase database of PPDB (Ganitkevitch et al., 2013) for feature extraction from arbitrary phrases that are simply $n$-grams. The second approach, on the other hand, needs reliable parsers or chunkers to identify phrase boundaries. Phrase alignment methods proposed by Ouyang and McKeown (2019) uses chunkers while the method proposed by Sultan et al. (2014) and Arase and Tsujii (2017) depend on the syntactic parser to obtain phrase structures. Although these lexical and paraphrase dictionaries, chunkers, and syntactic parsers are useful resources to realize high-quality phrase alignment, they restrict the applicability of phrase alignment methods. Because these resources assume to be applied to problems in the general domain, their performances are likely degraded in domain-specific areas. Besides, such resources are unlikely available other than in English. Different from these previous methods, SAPPHIRE is easily adaptable to any domains or languages because it requires only a raw corpus to train word embedding models. Furthermore, SAPPHIRE can handle both types of phrases with or without grammatical constraints.

Bilingual phrase pair extraction is a common technique in SMT. The typical approach is first obtaining word alignments by GIZA++ (Och and Ney, 2003) and then composing phrase alignment pairs. Different from the monolingual setting, bilingual word alignment can assume that an abundant parallel corpus is available. SAPPHIRE, on the other hand, requires only a raw corpus to train word embeddings, which is far easier to collect than monolingual parallel (*i.e.*, paraphrase) corpora.

## 5. Conclusion

We proposed SAPPHIRE, a simple phrase aligner that depends only on a monolingual corpus. Experiment results showed that SAPPHIRE outperformed the previous method and achieved the state-of-the-art phrase alignment F-measure score on the MSR RTE corpus. Our experiments also investigated in detail the effects of word embedding models and word alignment methods in SAPPHIRE.

As future works, we will apply SAPPHIRE to various domains and languages. Also, we will extend SAPPHIRE to identify grammatical phrase alignments without syntactic parsers by utilizing powerful pre-trained language models.

## Acknowledgments

This work was supported by JST, ACT-I, Grant Number JPMJPR16U2, Japan.

## Bibliographical References

Arase, Y. and Tsujii, J. (2017). Monolingual Phrase Alignment on Parse Forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–11.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., and Giampiccolo, D. (2006). The second PASCAL recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association of Computational Linguistics (TACL)*, 5:135–146.

Bourgeois, F. and Lassalle, J.-C. (1971). An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. *Communications of the Association for Computing Machinery (Commun. ACM)*, 14(12):802–804.

Brockett, C. (2007). Aligning the RTE 2006 Corpus. Technical report, Microsoft Research.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL Recognising Textual EntailmentChallenge. In *Proceedings of the First PASCAL Challenges Workshop on RTE*, pages 177–190.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 55–65.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The Paraphrase Database. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 758–764.

Kajiwara, T. and Komachi, M. (2016). Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1147–1158.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 127–133.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180.

Lan, W. and Xu, W. (2018). Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering.

In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 3890–3902.

Li, Z., Jiang, X., Shang, L., and Liu, Q. (2019). Decomposable Neural Paraphrase Generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3403–3414.

MacCartney, B., Galley, M., and Manning, C. D. (2008). A Phrase-Based Alignment Model for Natural Language Inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 802–811.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the Association for Computing Machinery (Commun. ACM)*, 38(11):39–41.

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

Ouyang, J. and McKeown, K. (2019). Neural Network Alignment for Sentential Paraphrases. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4724–4735.

Song, Y. and Roth, D. (2015). Unsupervised Sparse Vector Densification for Short Text Similarity. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1275–1280.

Sultan, M. A., Bethard, S., and Sumner, T. (2014). DLS@CU: Sentence Similarity from Word Alignment. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 241–246.

Thadani, K. and McKeown, K. (2011). Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 254–259.

Thadani, K., Martin, S., and White, M. (2012). A Joint Phrasal and Dependency Model for Paraphrase Alignment. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1229–1238.

Yao, X., Van Durme, B., Callison-Burch, C., and Clark, P. (2013). Semi-Markov Phrase-Based Monolingual Alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 590–600.