Monolingual Phrase Alignment on Parse Forests

Yuki Arase\textsuperscript{1*} and Junichi Tsujii\textsuperscript{2*}
\textsuperscript{1}Osaka University, Japan
\textsuperscript{2}Artificial Intelligence Research Center (AIRC), AIST, Japan

\textsuperscript{2}NaCTeM, School of Computer Science, University of Manchester, UK
arase@ist.osaka-u.ac.jp, j-tsujii@aist.go.jp

Abstract

We propose an efficient method to conduct phrase alignment on parse forests for paraphrase detection. Unlike previous studies, our method identifies syntactic paraphrases under linguistically motivated grammar. In addition, it allows phrases to non-compositionally align to handle paraphrases with non-homographic phrase correspondences. A dataset that provides gold parse trees and their phrase alignments is created. The experimental results confirm that the proposed method conducts highly accurate phrase alignment compared to human performance.

1 Introduction

Paraphrase detection is crucial in various applications, which has been actively studied for years. Due to difficulties caused by the non-homographic nature of phrase correspondences, the units of correspondence in previous studies are defined as sequences of words like in (Yao et al., 2013) and not syntactic phrases. On the other hand, syntactic structures are important in modeling sentences, e.g., their sentiments and semantic similarities (Socher et al., 2013; Tai et al., 2015).

In this paper, we present an algorithm to align syntactic phrases in a paraphrased pair of sentences. We show that (1) the problem of identifying a legitimate set of syntactic paraphrases under linguistically motivated grammar is formalized, (2) dynamic programming a la CKY (Cocke, 1969; Kasami, 1965; Younger, 1967) makes phrase alignment computationally feasible, (3) alignment quality of phrases can be improved using $n$-best parse forests instead of 1-best trees, and (4) non-compositional alignment allows non-homographic correspondences of phrases. Motivated by recent findings that syntax is important for phrase embedding (Socher et al., 2013) in which phrasal paraphrases allow semantic similarity to be replicated (Wieting et al., 2016, 2015), we focus on the syntactic paraphrase alignment.

Fig. 1 shows a real example of phrase alignments produced by our method. Alignment proceeds in a bottom-up manner using the compositional nature of phrase alignments. First, word alignments are given. Then, phrase alignments are recursively identified by supporting relations between phrase pairs. Non-compositional alignment is triggered when the compositionality is violated, which is common in paraphrasing.

For systematic research on syntactic phrase alignment in paraphrases, we constructed a gold standard dataset of paraphrase sentences with phrase alignment (20,678 phrases in 201 paraphrasal sentences). This dataset will be made public for future research on paraphrase alignment. The experimental results show that our method achieves 83.64\% and 78.91\% in recall and precision in terms of alignment pairs, which are 92\% and 89\% of human performance, respectively.
2 Related Work

Due to the large amount of sentence-level paraphrases collected (Dolan et al., 2004; Cohn et al., 2008; Heilman and Smith, 2010; Yin and Schütze, 2015; Biran et al., 2016), researchers can identify phrasal correspondences for natural language inferences (MacCartney et al., 2008; Thadani et al., 2012; Yao et al., 2013). Current methods extend word alignments to phrases in accordance with the methods in statistical machine translation. However, phrases are defined as a simple sequence of words, which do not conform to syntactic phrases. PPDB (Ganitkevitch et al., 2013) provides syntactic paraphrases similar to synchronous context free grammar (SCFG). As discussed below, SCCFG captures only a fraction of paraphrasing phenomenon.

In terms of our approach, parallel parsing is a relevant area. Smith and Smith (2004) related monolingual parses in different languages using word alignments, while Burkett and Klein (2008) employed phrase alignments. Moreover, Das and Smith (2009) proposed a model that generates a paraphrase of a given sentence using quasi-synchronous dependency grammar (Smith and Eisner, 2006). Since they used phrase alignments simply as features, there is no guarantee that the output alignments are legitimate.

Synchronous rewriting in parallel parsing (Kacshammer, 2013; Maillé de Buy Wenniger and Sima'an, 2013) derives parse trees that conform to discontinuous word alignments. In contrast, our method respects parse trees derived by linguistically motivated grammar while handling non-monotonic phrase alignment.

The synchronous assumption in parallel parsing has been argued to be too rigid to handle parallel sentence pairs or even paraphrased sentence pairs. Burkett et al. (2010) proposed weakly synchronized parallel parsing to tackle this problem. Although this model increases the flexibility, the obtainable alignments are restricted to conform to inversion transduction grammar (ITG) (Wu, 1997). Similarly, Choe and McClosky (2015) used dependency forests of paraphrasal sentence pairs and allowed disagreements to some extent. However, alignment quality was beyond their scope. Weese et al. (2014) extracted SCFG from paraphrase corpora. They showed that parsing was only successful in 9.1% of paraphrases, confirming that a significant amount of transformations in paraphrases do not conform to compositionality or ITG.

### 3 Formulation of Phrase Alignment

In this study, we formalize the problem of legitimate phrase alignment. For simplicity, we discuss tree alignment instead of forests using Fig. 2 as a running example.

#### 3.1 Notation

Table 1 describes the notation used in this paper. We call a paraphrased pair source sentence $s$ and the other as target $t$. Superscripts of $s$ and $t$ represent the source and the target, respectively. Specifically, $\langle \tau^s, \tau^t \rangle$ is a pair of source and target phrases. We represent $f_1/f_2/\cdots/f_i(\cdot)$ to abbreviate $f_i(\cdots f_2(f_1(\cdot)) \cdots )$ as an intuitive illustration. It should be noted that the order of the function symbols is reversed, e.g., $l/r(\tau) = r(l(\tau))$ derives the right-child of the left-child node of $\tau$, and $l/ds(\tau)$ derives the left descendants of $\tau$.

### 3.2 Definition of a Legitimate Alignment

A possible parse tree alignment of $s$ and $t$ is represented as a set of aligned pairs of phrases $\{(\tau^s_i, \tau^t_i)\}$. $\tau^s_i$ and $\tau^t_i$ are the source and the target phrases that constitute the $i$-th alignment, respectively. Either $\tau^s_i$ or $\tau^t_i$ can be $\tau_0$ when a phrase does not correspond to another sentence, which is called a null-alignment. Each phrase alignment can have support relations as:

**Definition 3.1.** A pair $h_i = (\tau^s_i, \tau^t_i)$ is supported by alignments of their descendant phrases when
Figure 2: Alignment pair and its supports

\[ h_i = (\tau_i^s, \tau_i^t) \]

\[ \langle l/ds(\tau_i^s), r/ds(\tau_i^t) \rangle \]

or \[ \langle l/ds(\tau_i^s), r/ds(\tau_i^t) \rangle \]

exists. Pre-terminal phrases are supported by the corresponding word alignments.

Support relations are denoted using \( \Rightarrow \) or \( \not\Rightarrow \) that represent the order of support phrases. Specifically, \( \langle l/ds(\tau_i^s), l(d(\tau_i^t)) \rangle \Rightarrow h_i \) is straight while \( \langle l/ds(\tau_i^s), r(d(\tau_i^t)) \rangle \not\Rightarrow h_i \) is inverted. In Fig. 2, \( \langle l/\tau_i^s, m/\tau_i^m \rangle, \langle n/\tau_i^n \rangle \Rightarrow h_i \), where \( \tau_m = l/ds(\tau_i^s) \) and \( \tau_n = r/ds(\tau_i^t) \).

The number of all possible alignments in \( s \) and \( t \), which is denoted as \( H \), is exponential to the length. However, only its fraction constitutes legitimate parse tree alignments. For example, a subset in which the same phrase in \( s \) is aligned with multiple phrases in \( t \), called competing alignments, is not legitimate as a parse tree alignment. The relationships among phrases in parse trees impose constraints on a subset to provide legitimacy.

Given word alignments \( W \) that provide the basis for the phrase alignment, its legitimate set \( W_L \subset W \) should be 1-to-1 alignments. Starting with \( W_L \), a legitimate set of phrase alignments \( H_L(\subset H) \) with an accompanying set of support relations, \( H_L(\subset \Delta) \) is constructed. A legitimate set of alignments \( \langle H_L, \Delta_L \rangle \) can be enlarged only by adding \( h_i \) to \( H_L \) with either the support relation \( \Rightarrow \) or \( \not\Rightarrow \) added to \( \Delta_L \). These assume competing alignments among the child phrases, thus cannot co-exist in the same legitimate set.

\( h_i \) can be supported by more than one pair of descendant alignments in \( \Delta_L \), i.e., \( \{ h_i \} \Rightarrow h_i \) or \( \not\Rightarrow \) \( h_i \) exists. For \( H_m = \{ h_m \} \), we define the relationship \( \leq \) for alignments, i.e., \( h_p \leq h_q \) meaning that \( \tau_p^s \in ds(\tau_q^s) \) \( \wedge \tau_p^t \in ds(\tau_q^t) \).

For example, in Fig. 2, \( h_m \leq h_i \) and \( h_n \leq h_i \).

**Theorem 3.1.** There always exist the maximum pair \( h_M \in H_m \) where \( \forall h_m \in H_m, h_m \leq h_M \).

\( \langle H_L, \Delta_L \rangle \) should satisfy the conditions in Definition 3.2 to be legitimate as a whole. We denote \( h_i \Rightarrow h_j \) when a chain exists in \( \Delta_L \), which connects \( h_i \) to \( h_j \) regardless of straight or inverted directions of intermediate supports, e.g., \( \{ h_i, \cdot \} \Rightarrow h_{i+1}, \{ h_{i+1}, \cdot \} \not\Rightarrow h_{i+2}, \ldots, \{ h_{j-1}, \cdot \} \Rightarrow h_j \).

**Definition 3.2.** \( \langle H_L, \Delta_L \rangle \) should satisfy:

1. **Root-Pair Containment:** \( \langle \tau_R^s, \tau_R^t \rangle \in H_L \)

2. **Same-Tree:** \( \{ \tau_i^s \mid \langle \tau_i^s, \tau_i^t \rangle \in H_L \} \) are subsets of phrases in the same complete parse tree of \( s \) (same for \( t \)).

3. **Relevance:** \( \forall h_i \in H_L, h_i \Rightarrow \langle \tau_R^s, \tau_R^t \rangle \in \Delta_L \)

4. **Consistency:** In \( H_L \), a phrase \( \not\in \tau_0 \) in the source tree is aligned with at most one phrase \( \not\in \tau_0 \) in the target tree, and vice versa.

5. **Monotonous:** For \( \langle \tau_i^s, \tau_j^t \rangle \in H_L \), \( \tau_i^s \in ds(\tau_j^t) \) if \( \tau_j^t \in ds(\tau_i^s) \).

6. **Maximum Set:** \( H_L \) is the maximum legitimate set, in the sense that \( \forall \langle \tau^s, \tau^t \rangle \in (H \setminus H_L), \{ \langle \tau^s, \tau^t \rangle \} \cup H_L \) cannot be a legitimate set with any \( \Delta \).

The **Same-Tree** condition is required to conduct an alignment on forests that consist of multiple trees in a packed representation. The **Consistency** condition excludes competing alignments. The **Monotonous** condition is a consequence of compositionality. The **Maximum Set** means if \( h_m, h_n \in H_L \) are in positions of a parse tree that can support \( h_i \), \( h_l \) and the support relation should be added to \( \langle H_L, \Delta_L \rangle \). Such a strict locality of compositionality is often violated in practice as discussed in Sec. 2. To tackle this issue, we add another operation to align phrases in a non-compositional way in Sec. 4.3.

### 3.3 Lowest Common Ancestor

The same aligned pair can have more than one support of descendant alignments because there are numerous descendant node combinations. However, the **Monotonous** and the **Maximum Set** conditions allow \( \Delta_L \) to be further restricted so that each of aligned pairs in \( H_L \) has only one support.

Let us assume that alignment \( h_i \) is supported by more than one pair of descendant alignments
in \( \Delta_L \), i.e., \( \Delta_L \supseteq \{ \{ h_m, h_n \} \Rightarrow h_i \} \). We denote \( H_m = \{ h_m \} \) and \( H_n = \{ h_n \} \). For each \( h_m \in H_m \) and \( h_n \in H_n \), we remove all support relations from \( \Delta_L \) except for the maximum pairs or the pre-terminal alignments. The resultant set \( \Delta_L \) satisfies:

**Theorem 3.2.** For all \( \{ h_m, h_n \} \Rightarrow h_i \) \( \in \Delta_L \), \( \tau_i^t = lca(\tau_m^t, \tau_n^t) \) and \( \tau_i^s = lca(\tau_m^s, \tau_n^s) \) are true.

In Fig. 2, \( \tau_i^s \) is the lowest common ancestor (LCA) of \( \tau_m^s \) and \( \tau_n^s \), and \( \tau_i^t \) is the LCA of \( \tau_m^t \) and \( \tau_n^t \). Theorem 3.2 constitutes the basis for the dynamic programming (DP) in our phrase alignment algorithm (Sec. 4.2).

## 4 Modeling of Phrase Alignment

We formally model the phrase alignment process as illustrated in Fig. 3, where \( h_i \) is aligned from descendant alignments, i.e., \( h_m \) and \( h_n \).

### 4.1 Probabilistic Model

Similar to the probabilistic context free grammar (PCFG), the inside probability \( \alpha_i \) of \( h_i \) is determined by the inside probabilities, \( \alpha_m \) and \( \alpha_n \), of the support pairs, together with the probability of the rule, i.e., the way by which \( h_m \) and \( h_n \) are combined to support \( h_i \) as shown in Fig. 3. It is characterized by four paths, \( \pi_{m,i} \) (the path from \( \tau_m^t \) to \( \tau_i^t \)), \( \pi_{m,i}^t \) (\( \tau_m^s \) to \( \tau_i^s \)), \( \pi_{n,i}^t \) (\( \tau_m^t \) to \( \tau_i^t \)), and \( \pi_{n,i}^s \) (\( \tau_n^s \) to \( \tau_i^s \)).

Each path consists of a set of null-aligned phrases \( \phi \in \{ \phi, \tau_0 \} \) and their mothers, e.g., the path \( \pi_{m,i} \) in Fig. 3 is a set of \( \{ \phi_1^t, m(\phi_1^t) \} \), \( \{ \phi_2^t, m(\phi_2^t) \} \), and \( \{ \phi_3^t, m(\phi_3^t) \} \). We assume that each occurrence of a null-alignment is indepen-

---

1 Theorem 3.2. For all \( \{ h_m, h_n \} \Rightarrow h_i \) \( \in \Delta_L \), \( \tau_i^t = lca(\tau_m^t, \tau_n^t) \) and \( \tau_i^s = lca(\tau_m^s, \tau_n^s) \) are true.

Figure 3: Inside probability depends on support alignments and paths to reach an LCA.

Figure 4: Alignment pairs and packed supports.
Algorithm 4.1 Phrase Alignment

1: LCAs and $\gamma$ in parse trees of $s$ and $t$ are computed and stored in $Lca^s[]$ and $Lca^t[]$.
2: set $A[\tau^s] \leftarrow \emptyset$ for all $\tau^s$.
3: for all $(w^s, w^t) \in W$ do
4: find $\tau^s$ and $\tau^t$ covering $w^s$ and $w^t$.
5: Compute $\alpha_i$ of $\langle \tau^s, \tau^t \rangle$ using Eq. (1).
6: PACK$(\tau^s, \tau^t), (\alpha_i, (\emptyset, A))$
7: for all $\tau^s_m, \tau^t_n$ do $\triangleright$ Trace the source tree from the bottom to top.
8: for all $(\langle \tau^s_m, \gamma^s_{m,n,i} \rangle) \subseteq Lca^s[\tau^s_m] \cap A[\tau^s]$ do
9: ALIGN$(\tau^s_m, \tau^s_i, \gamma^s_{m,n,i}, A)$
10: function ALIGN$(\tau^s, \tau^t, \gamma^s, A)$
11: for all $\ll_i = (\langle \tau^s_i, \tau^t_i \rangle) \subseteq A[\tau^t]$ do
12: for all $\ll_i = (\langle \tau^s_i, \tau^t_i \rangle) \subseteq A[\tau^t]$ do
13: $(\langle \tau^s_i, \alpha \gamma^s \rangle) \leftarrow Lca^s[\tau^s_i] \cap A[\tau^s]$.
14: Compute $\alpha_i$ using Eq. (1).
15: PACK$(\tau^s_i, \tau^t_i), (\alpha_i, (\emptyset, A))$
16: function PACK$(\tau^s, \tau^t), (\alpha, (\emptyset, A))$
17: if $\langle \tau^s, \tau^t \rangle \in A[\tau^s]$ then
18: $A[\tau^s] \leftarrow A[\tau^s] \cup (\alpha, (\emptyset, A))$ $\triangleright$ Merge supports and their inside probability
19: else
20: $A[\tau^s] \leftarrow (\langle \tau^s, \tau^t \rangle, (\alpha, (\emptyset, A)))$

support relations are packed as a support list$^2$ by the PACK function.

4.3 Non-Compositional Alignment

A monotonic alignment requires $\tau^s_i \in \ll_i$ and $\tau^t_i \in \ll_i$ to have an LCA, which adheres to the compositionality in language. However, previous studies declared that the compositionality is violated in a monolingual phrase alignment (Burkett et al., 2010; Weese et al., 2014). Heilman and Smith (2010) discuss complex phrase reordering is prevalent in paraphrases and entailed text.

A non-monotonic alignment occurs when corresponding phrases have largely different orders, i.e., one of them (e.g., $\tau^s_m$) is an ancestor of another (e.g., $\tau^t_n$) of the same phrase. Such a case could be exceptionally compatible, when $\tau^t_n$ has null-alignments and all the aligned phrases of $\tau^s_m$ fit in these null-alignments. A new alignment $\langle \tau^s_m, \tau^t_n \rangle (= \tau^s_m) \triangleright$ would be non-monotonically formed. Fig. 5 shows a real example of non-compositional alignment produced by our method. The target phrase $\tau^s_m$ (“through the spirit of teamwork”) is null-alignment when aligning $\tau^s_m$ and $\tau^t_n$, but then the alignment to $\tau^s_m$ (“Relying on team spirit”) is allowed by non-compositional alignment of $\tau^s_m$.

Unlike monotonous alignment, we have to verify whether the internal structures of $\tau^s_m$ and $\tau^t_n$ are compatible. Since the internal structures of $\tau^s_m$ and $\tau^t_n$ depend on their supporting alignments, their packed representations in $A$ have to be unpacked, and each pair of supporting alignments for $\ll_i$ and $\ll_i$ must be checked to confirm compatibility. Furthermore, since the aligned phrases inside $\tau^s_m$ and $\tau^t_n$ have their own null-alignments, we need to unpack deeper supporting alignments as well.

Algorithm 4.2 checks if target phrases $\tau^s_m$ and $\tau^t_n \in ds(\tau^s_m)$ are compatible. We use the following notations: $\langle \tau^s_m \rangle$ and $\langle \tau^t_n \rangle$ represent the phrases of $\tau^s_m$ and $\tau^t_n$ with the $i$-th and $j$-th sets of supporting alignments, respectively. For $\langle \tau^s_m \rangle$ in Fig. 4, there are $[\tau^s_m]_{1} \subseteq \ll_i$ supported by $\ll_j$ and $[\tau^s_m]_{2} \subseteq \ll_i$ supported by $\ll_j$. $[\tau^s_m]_{1}$ consists of sets of aligned target phrases $\psi(\tau^s_m) = \Psi_k^{[\tau^s_m]} \cap \Phi_k^{[\tau^s_m]}$ and null-alignments $\Phi_k^{[\tau^s_m]} = \Phi_j^{[\tau^s_m]}$ if $\tau^s_m$ is similar.

For each $\langle \tau^s_m \rangle$, if $\tau^s_m$ fits in its null-alignment like in Fig. 5, the alignment information is updated at line 5, where GAP function takes two phrases and returns a set of null-alignments on a path between them. If $\tau^s_m$ is a descendant of a support of $\tau^s_m$, the compatibility is recursively checked (line 7). Otherwise, the compatibility of the supports of $\tau^s_m$ and $\tau^t_n$ are recursively checked in DOWN function in a similar manner (line 10).

When Trace function returns a set of $\{\psi_k, \Phi_k\}$, all $\psi \in \Psi_k$ are aligned with phrases in the source and their inside probabilities are stored in $A$. Thus we can compute the inside probability for each $\langle \Psi_k, \Phi_k \rangle$, which is stored in $A$ to-

$^2$This is true except for a non-compositional alignment where the packed representation must be unpacked.
Source: Relying on team spirit, expedition members defeated difficulties.

Target: Members of the scientific team overcame difficulties through the spirit of teamwork.

Figure 5: Example of a non-compositional alignment

together with a new alignment pair \((\tau^s_i, \tau^t_i)\) where 
\(\tau^s_i = lca(\tau^s_m, \tau^s_n)\) and \(\tau^t_i = \tau^t_m\).

4.4 Forest Alignment

Although we have discussed using trees for clarity, the alignment is conducted on forests. The alignment process is basically the same. The only difference is that the same pair has multiple LCAs. Hence, we need to verify if the sub-trees can be on the same tree when identifying their LCAs since multiple nodes may cover the same span with different derivations. This is critical for non-compositional alignment because whether the internal structures are on the same tree must be confirmed while unpacking them.

Our alignment process corresponds to re-ranking of forests and may derive a different tree from the 1-best, which may resolve ambiguity in parsing. We use a parser trained beforehand because joint parsing and alignment is computationally too expensive.

5 Parameterization

Next, we parameterize the alignment probability.

5.1 Feature-enhanced EM Algorithm

We apply the feature-enhanced EM (Berg-Kirkpatrick et al., 2010) due to its ability to use dependent features without an irrational independence assumption. This is preferable because the attributes of phrases largely depend on each other.

Our method is computationally heavy since it handles forests and involves unpacking in the non-compositional alignment process. Thus, we use Viterbi training (Brown et al., 1993) together with a beam search of size \(\mu_b \in \mathbb{N}\) on the feature-enhanced EM. Also, mini-batch training (Liang and Klein, 2009) is applied. Such an approximation for efficiency is common in parallel parsing (Burkett and Klein, 2008; Burkett et al., 2010).

In addition, an alignment supported by distant descendants tends to fail to reach a root-pair alignment. Thus, we restrict the generation gap between a support alignment and its LCA to be less than or equal to \(\mu_g \in \mathbb{N}\).

5.2 Features

In feature-enhanced EM, the alignment probability in Eq. (1) is parameterized using features:

\[
P_r(\tau^s_i, \tau^t_i) \propto \frac{\exp(w \cdot F(a^s_i, a^t_j))}{\sum_{(\tau^s_j, \tau^t_j), \tau^t_j = \tau^t_m} \exp(w \cdot F(a^s_j, a^t_j))},
\]

where \(a = (a_0, \ldots, a_n)\) consists of \(n\) attributes of \(\tau\). \(F(\cdot, \cdot)\) and \(w\) are vectors of feature functions and their weights, respectively.

In a parse tree, the head of a phrase determines its property. Hence, a lemmatized lexical head \(a_{\text{lex}} \in a\) combined with its syntactic category \(a_{\text{cat}} \in a\) is encoded as a feature\(^3\) as shown below. We use semantic (instead of syntactic) heads to encode semantic relationships in paraphrases.

1: \(1(a^s_{\text{lex}} = \cdot, a^s_{\text{cat}} = \cdot, a^t_{\text{lex}} = \cdot, a^t_{\text{cat}} = \cdot)\)
2: \(1(\text{SurfaceSim}(a^s_{\text{lex}} = \cdot, a^t_{\text{lex}} = \cdot))\)
3: \(1(\text{WordnetSim}(a^s_{\text{lex}} = \cdot, a^t_{\text{lex}} = \cdot))\)
4: \(1(\text{EmbeddingSim}(a^s_{\text{lex}} = \cdot, a^t_{\text{lex}} = \cdot))\)
5: \(1(\text{IsPrepositionPair}(a^s_{\text{lex}} = \cdot, a^t_{\text{lex}} = \cdot))\)
6: \(1(a^s_{\text{cat}} = \cdot, a^t_{\text{cat}} = \cdot)\)
7: \(1(\text{IsSameCategory}(a^s_{\text{cat}} = \cdot, a^t_{\text{cat}} = \cdot))\)

The first feature is an indicator invoked only at specific values. On the other hand, the rest of the

\(^3\)We also tried features based on the configurations of the source and target sub-trees similar to (Das and Smith, 2009) as well as features based on the spans of null-alignments. However, none of them contributed to alignment quality.
features are invoked across multiple values, allowing general patterns to be learned. The second feature is invoked if two heads are identical or a head is a substring of another. The third feature is invoked if two heads are synonyms or derivations that are extracted from the WordNet\textsuperscript{4}. The fourth feature is invoked if the cosine similarity between word embeddings of two heads is larger than a threshold. The fifth feature is invoked when the heads are both prepositions to capture their different natures from the content words. The last two features are for categories; the sixth one is invoked if the input categories are the same.

To avoid generating a huge number of features, we reduce the number of syntactic categories; for contents (N, V, ADJ, and ADV), prepositions, coordinations, null (i.e., for $\tau_0$), and others.

5.3 Penalty Function
Since our method allows null-alignments, it has a degenerate maximum likelihood solution (Liang and Klein, 2009) that makes every phrase null-alignment. Similarly, a degenerate solution overly conducts non-compositional alignment.

To avoid these issues, a penalty is incorporated:

$$P_c(\tau^s_t, \tau^t_s) = \begin{cases} 
\exp\{- (|\tau^s_t|_\phi + |\tau^t_s|_\phi + \mu_c + 1)^{\mu_n} \} & \text{(non-compositional alignment)} \\
\exp\{- (|\tau^s_t|_\phi + |\tau^t_s|_\phi + 1)^{\mu_n} \} & \text{(otherwise)}
\end{cases}$$

where $| \cdot |_\phi$ computes the span of internal null-alignments, and $\mu_n \geq 1.0$ and $\mu_c \in \mathbb{R}_+$ control the strength of the penalties of the null-alignment and the non-compositional alignment, respectively. The penalty function is multiplied by Eq. (1) as a soft-constraint for re-ranking alignment pairs in Algorithm 4.1.

5.4 Combination with Parse Probability
Following the spirit of parallel parsing that simultaneously parses and aligns sentences, we linearly interpolate the alignment probability with the parsing probability once the parameters are tuned by EM. When aligning a node pair $<\tau^s_t, \tau^t_s>$, the overall probability is computed as:

$$(1 - \mu_p)\alpha_i + \mu_p \phi(\tau^s_t) \phi(\tau^t_s),$$

where $\phi(\cdot)$ gives the marginal probability in parsing and $\mu_p \in [0, 1]$ balances these probabilities.

6 Evaluation
As discussed in Sec. 2, previous studies have not conducted syntactic phrase alignment on parse trees. A direct metric does not exist to compare paraphrases that cover different spans, i.e., our syntactic paraphrases and paraphrases of $n$-grams.

Thus, we compared the alignment quality to that of humans as a realistic way to evaluate the performance of our method.

We also evaluated the parsing quality. Similar to the alignment quality, differences in phrase structures disturb the comparisons (Sagae et al., 2008).

Our method applies an HPSG parser Enju (Miyao and Tsujii, 2008) to derive parse forests due to its state-of-the-art performance and ability to provide rich properties of phrases. Hence, we compared our parsing quality to the 1-best parses of Enju.

6.1 Language Resources
We used reference translations to evaluate machine translations\textsuperscript{5} as sentential paraphrases (Weese et al., 2014). The reference translations of 10 to 30 words were extracted and paired, giving $41K$ pairs as a training corpus.

We use different kinds of dictionaries to obtain word alignments W as well as to compute feature functions. First, we extract synonyms and words with derivational relationship using WordNet. Then we handcraft derivation rules (e.g., create, creation, creator) and extract potentially derivational words from the training corpus. Finally, we use prepositions defined in (Srikumar and Roth, 2013) as a preposition dictionary to compute the feature function.

In addition, we extend W using word embeddings; we use the MLVLSA word embeddings (Rastogi et al., 2015) given the superior performance in word similarity tasks. Specifically, we compute the cosine similarity of embeddings; words with a higher similarity value than a threshold are determined as similar words. The threshold is empirically set as the 100th highest similarity value between words in the training corpus.

6.2 Gold-Standard Data
Since no annotated corpus provides phrase alignments on parse trees, we created one through two-phase manual annotation. First, a linguistic expert with rich experience on annotating HPSG trees

\footnote{\textsuperscript{4}http://wordnet.princeton.edu}
annotated gold-trees to paraphrasal sentence pairs sampled from the training corpus. To diversify the data, only one reference pair per sentence of a source language was annotated. Consequently, 201 paraphrased pairs with gold-trees (containing 20,678 phrases) were obtained.

Next, three professional English translators identified paraphrased pairs including null-alignments given sets of phrases extracted from the gold-trees. These annotators independently annotated the same set, yielding 14,356 phrase alignments where at least one annotator regarded as a paraphrase. All the annotators agreed that 77% of the phrases were paraphrases.

We used 50 sentence pairs for development and another 151 for testing. These pairs were excluded from the training corpus.

6.3 Evaluation Metric

Alignment Quality Alignment quality was evaluated by measuring the extent that the automatic alignment results agree with those of humans. Specifically, we evaluated how gold-alignments can be replicated by automatic alignment (called recall) and how automatic alignments overlap with alignments that at least an annotator aligned (called precision) as:

\[
\text{Recall} = \frac{|\{h | h \in H_a \land h \in G \cap G'\}|}{|G \cap G'|},
\]

\[
\text{Precision} = \frac{|\{h | h \in H_a \land h \in G \cup G'\}|}{|H_a|},
\]

where \(H_a\) is a set of alignments, while \(G\) and \(G'\) are the ones that two of annotators produce, respectively. The function of \(|\cdot|\) counts the elements in a set. There are three combinations for \(G\) and \(G'\) because we had three annotators. The final precision and recall values are their averages.

Parsing Quality The parsing quality was evaluated using the CONLL-X (Buchholz and Marsi, 2006) standard. Dependencies were extracted from the output HPSG trees, and evaluated using the official script.\(^6\) Due to this conversion, the accuracy on the relation labels is less important. Thus, we reported only the unlabeled attachment score (UAS).\(^7\) The development and test sets provide 2,371 and 6,957 dependencies, respectively.

---

<table>
<thead>
<tr>
<th>Roles of hyper-parameters</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu_n) Control penalty for null-alignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mu_c) Control penalty for non-compositional alignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mu_p) Balance alignment and parsing prob.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mu_b) Beam size at alignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mu_g) Generation gap to reach an LCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Summary of the hyper-parameters

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Prec.</th>
<th>UAS</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>90.65</td>
<td>88.21</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Proposed</td>
<td>83.64</td>
<td>78.91</td>
<td>93.49</td>
<td>98</td>
</tr>
<tr>
<td>Monotonic</td>
<td>82.86*</td>
<td>77.97*</td>
<td>93.49</td>
<td>98</td>
</tr>
<tr>
<td>w/o EM</td>
<td>81.33*</td>
<td>75.09*</td>
<td>92.91*</td>
<td>86</td>
</tr>
<tr>
<td>1-best tree</td>
<td>80.11*</td>
<td>73.26*</td>
<td>93.56</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3: Evaluation results on the test set, where * represents p-value < 0.05 against our method.

Since all metrics were computed in a set, the approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) \((B = 10K)\) was used for significance testing. It has been shown to be more conservative than using bootstrap resampling (Riezler and Maxwell, 2005).

6.4 Results and Discussion

Overall Results Table 2 summarizes the hyper-parameters, which were tuned to maximize UAS in the development set using the Bayesian optimization. For efficiency, we used \(2K\) samples from the training corpus and set the mini-batch size in feature-enhanced EM to 200 similar to “rapid training” in (Burkett and Klein, 2008). We also set \(\mu_b = 50\) during EM training to manage the training time.

Table 3 shows the performance on the test set for variations of our method and that of the human annotators. The last column shows the percentage of pairs where a root pair is reached to be aligned, called reachability. Our method is denoted as Proposed, while its variations include a method with only monotonic alignment (monotonic), without EM (w/o EM), and a method aligning only 1-best trees (1-best tree).

The performance of the human annotators was assessed by considering one annotator as the test and the other two as the gold-standard, and then taking the averages, which is the same setting as our method. We regard this as the pseudo inter-
annotator agreement, since the conventional interannotator agreement is not directly applicable due to variations in aligned phrases.

Our method significantly outperforms the others as it achieved the highest recall and precision for alignment quality. Our recall and precision reach 92% and 89% of those of humans, respectively. Non-compositional alignment is shown to contribute to alignment quality, while the feature-enhanced EM is effective for both the alignment and parsing quality. Comparing our method and the one aligning only 1-best trees demonstrates that the alignment of parse forests largely contributes to the alignment quality. Although we confirmed that aligning larger forests slightly improved recall and precision, the improvements were statistically insignificant. The parsing quality was not much affected by phrase alignment, which is further investigated in the following.

Finally, our method achieved 98% reachability, where 2% of unreachable cases were due to the beam search. While understanding that the reachability depends on experimental data, ours is notably higher than that of SCFG, reported as 9.1% in (Weese et al., 2014). These results show the ability of our method to accurately align paraphrases with divergent phrase correspondences.

**Effect of Mini-Batch Size** We investigated the effect of the mini-batch size in EM training using the entire training corpus (41K pairs). When increasing the mini-batch size from 200 to 2K, recall, precision, and UAS values are fairly stable. In addition, they are insensitive against the amount of training corpus, showing the comparable values of unreachable cases were due to erroneous original sentences that should be disregarded, e.g., “For two weeks ago,...” and “Accord-
ing to the source, will also meet...”. Consequently, our method corrected 32 errors while introducing 23 errors in reality for 446 errors in 1-best trees, which achieves a 2.5% error reduction.

These are promising results for our method to improve parsing quality, especially on the PP-attachment (159 errors in 1-best), which contained 14 of the 32 corrected errors. Fig. 1 shows a real example; the phrase of “for a smoke” in the source was mistakenly attached to “ground floor” in the 1-best tree. This error was corrected as depicted.

Duan et al. (2016) showed that paraphrases artificially generated using n-best parses improved the parsing quality. One reason for limited improvement in our experiments may be because structural changes in our natural paraphrases are more dynamic than the level useful to resolve ambiguities. We will further investigate this in future.

7 Conclusion

We propose an efficient method for phrase alignment on parse forests of paraphrased sentences. To increase the amount of collected paraphrases, we plan to extend our method to align comparable paraphrases that are partially paraphrasal sentences. In addition, we will apply our method to parallel parsing and other grammar, e.g., projective dependency trees. Furthermore, we will apply such syntactic paraphrases to phrase embedding.

**Acknowledgments**

We thank Professor Issei Sato for permission to use their package for Bayesian optimization. Special thanks also to Dr. Yuka Tateishi for her contribution to HPSG tree annotation. Advice and comments given by Professor Takuya Matsuzaki and Professor Yusuke Miyao have been a great help in applying Enju parser for this project. We appreciate the anonymous reviewers for their insightful comments and suggestions to improve the paper. This project is funded by Microsoft Research Asia and the Kayamori Foundation of Informational Science Advancement.

**Supplemental Material**

The supplemental material is available at our web site\(^9\) that provides proofs of the theorems, pseudocodes of the algorithms, and more experiment results with examples.

\(^8\) Alignments were obtained by the model trained using the entire corpus with the 1K mini-batch size.

\(^9\)http://www-bigdata.ist.osaka-u.ac.jp/arase/pj/phrase-alignment/
References


