Monolingual Phrase Alignment on Parse Forests

Yuki Arase[1⋆] and **Junichi Tsujii**[⋆2]
[1]Osaka University, Japan
[⋆]Artificial Intelligence Research Center (AIRC), AIST, Japan
[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 programing 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.

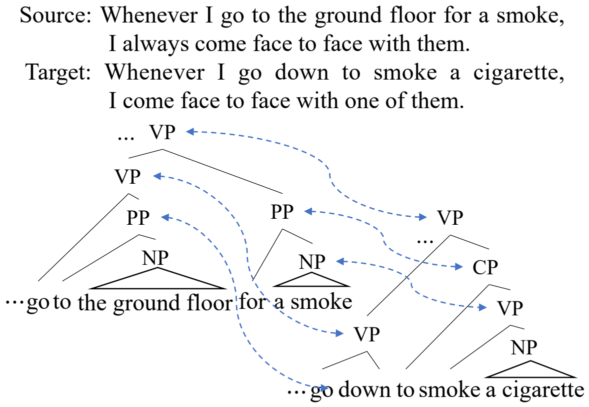
Source: Whenever I go to the ground floor for a smoke,
I always come face to face with them.
Target: Whenever I go down to smoke a cigarette,
I come face to face with one of them.
... VP
VP
PP
NP
PP
NP
···go to the ground floor for a smoke
VP
...
CP
VP
VP
NP
···go down to smoke a cigarette

Figure 1: Example of phrase alignments

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 experiment 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.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1–11
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

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, SCFG 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 (Kaeshammer, 2013; Maillette 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 paraphrasal 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.

	Explanation
s, t	Source and target sentences
τ	Phrase in the parse tree
$\tau_R, \tau_\emptyset$	τ_R is a phrase of a root node; $\tau_\emptyset$ is a special phrase with the null span that exists in every parse tree
ϕ	Phrase aligned to $\tau_\emptyset$
$\langle \cdot, \cdot \rangle$	Pair of entities; a pair itself can be regarded as an entity
$\{\cdot\}$	Set of entities
$m(\cdot)$	Derive the mother node of a phrase
$l(\cdot)$, $r(\cdot)$	Derive the left and right child nodes, respectively
$ds(\cdot)$	Derive descendants of a node including self; $\tau \in ds(\tau)$
$lca(\cdot, \cdot)$	Derive the lowest common ancestor (LCA) of two phrases

Table 1: Notation summary

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 τ, and $l/ds(\tau)$ derives the left descendants of τ.

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 $\{\langle \tau^s_i, \tau^t_i \rangle\}$. τ^s_i and τ^t_i are the source and the target phrases that constitute the i-th alignment, respectively. Either τ^s_i or τ^t_i can be $\tau_\emptyset$ 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* $\mathbb{h}_i = \langle \tau^s_i, \tau^t_i \rangle$ *is supported by alignments of their descendant phrases when*

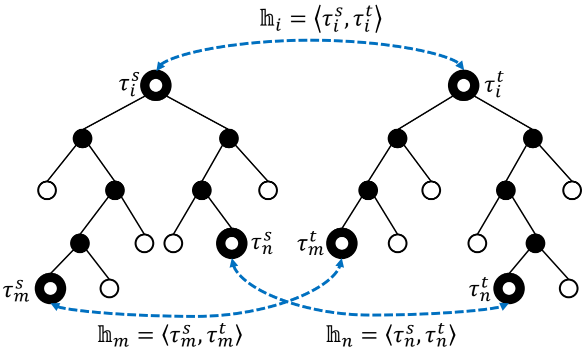
$\mathbb{h}_i = \langle \tau_i^s, \tau_i^t \rangle$
τ_i^s
τ_i^t
τ_n^s
τ_m^t
τ_m^s
τ_n^t
$\mathbb{h}_m = \langle \tau_m^s, \tau_m^t \rangle$
$\mathbb{h}_n = \langle \tau_n^s, \tau_n^t \rangle$

Figure 2: Alignment pair and its supports

$\langle\langle l/ds(\tau_i^s), l/ds(\tau_i^t)\rangle, \langle r/ds(\tau_i^s), r/ds(\tau_i^t)\rangle\rangle$ or $\langle\langle l/ds(\tau_i^s), r/ds(\tau_i^t)\rangle, \langle r/ds(\tau_i^s), l/ds(\tau_i^t)\rangle\rangle$ exists. Pre-terminal phrases are supported by the corresponding word alignments.

Support relations are denoted using $\Rightarrow$ or $\overset{R}{\Rightarrow}$ that represent the order of support phrases. Specifically, $\langle\langle l(\tau_i^s), l(\tau_i^t)\rangle, \langle r(\tau_i^s), r(\tau_i^t)\rangle\rangle \Rightarrow \mathbb{h}_i$ is straight while $\langle\langle l(\tau_i^s), r(\tau_i^t)\rangle, \langle r(\tau_i^s), l(\tau_i^t)\rangle\rangle \overset{R}{\Rightarrow} \mathbb{h}_i$ is inverted. In Fig. 2, $\langle\langle \tau_m^s, \tau_m^t\rangle, \langle \tau_n^s, \tau_n^t\rangle\rangle \Rightarrow \mathbb{h}_i$, where $\tau_m^s = l/ds(\tau_i^s)$ and $\tau_n^s = r/ds(\tau_i^s)$.

The number of all possible alignments in s and t, which is denoted as $\mathbb{H}$, 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 $\mathbb{W}$ that provide the basis for the phrase alignment, its legitimate set $\mathbb{W}_L \subset \mathbb{W}$ should be 1-to-1 alignments. Starting with $\mathbb{W}_L$, a legitimate set of phrase alignments $\mathbb{H}_L (\subset \mathbb{H})$ with an accompanying set of support relations, $\Delta_L (\subset \Delta)$ is constructed. A legitimate set of alignments $\langle \mathbb{H}_L, \Delta_L\rangle$ can be enlarged only by adding $\mathbb{h}_i$ to $\mathbb{H}_L$ with either the support relation $\Rightarrow$ or $\overset{R}{\Rightarrow}$ added to Δ_L. These assume competing alignments among the child phrases, thus cannot co-exist in the same legitimate set.

$\mathbb{h}_i$ can be supported by more than one pair of descendant alignments in Δ_L, *i.e.*, $\{\langle \mathbb{h}_m, \cdot\rangle\} \Rightarrow \mathbb{h}_i$ or $\{\langle \mathbb{h}_m, \cdot\rangle\} \overset{R}{\Rightarrow} \mathbb{h}_i$ exists. For $\mathbb{H}_m = \{\mathbb{h}_m\}$, we define the relationship $\leq$ for alignments, *i.e.*, $\mathbb{h}_p \leq \mathbb{h}_q$ meaning that $\tau_p^s \in ds(\tau_q^s) \wedge \tau_p^t \in ds(\tau_q^t)$. For example, in Fig. 2, $\mathbb{h}_m \leq \mathbb{h}_i$ and $\mathbb{h}_n \leq \mathbb{h}_i$.

Theorem 3.1. *There always exist the maximum pair $\mathbb{h}_M \in \mathbb{H}_m$ where $\forall \mathbb{h}_m \in \mathbb{H}_m, \mathbb{h}_m \leq \mathbb{h}_M$.*

$\langle \mathbb{H}_L, \Delta_L\rangle$ should satisfy the conditions in Definition 3.2 to be legitimate as a whole. We denote $\mathbb{h}_i \overset{*}{\mapsto} \mathbb{h}_j$ when a chain exists in Δ_L, which connects $\mathbb{h}_i$ to $\mathbb{h}_j$ regardless of straight or inverted directions of intermediate supports, e.g., $(\langle \mathbb{h}_i, \cdot\rangle \Rightarrow \mathbb{h}_{i+1}), (\langle \mathbb{h}_{i+1}, \cdot\rangle \overset{R}{\Rightarrow} \mathbb{h}_{i+2}), \ldots, (\langle \mathbb{h}_{j-1}, \cdot\rangle \Rightarrow \mathbb{h}_j)$. Note $\mathbb{h}_i \overset{*}{\mapsto} \mathbb{h}_i$ is always true.

Definition 3.2. *$\langle \mathbb{H}_L, \Delta_L\rangle$ should satisfy:*

1. *Root-Pair Containment: $\langle \tau_R^s, \tau_R^t\rangle \in \mathbb{H}_L$*

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

3. *Relevance: $\forall \mathbb{h}_i \in \mathbb{H}_L, \mathbb{h}_i \overset{*}{\mapsto} \langle \tau_R^s, \tau_R^t\rangle \in \Delta_L$*

4. *Consistency: In $\mathbb{H}_L$, a phrase ($\neq \tau_\emptyset$) in the source tree is aligned with at most one phrase ($\neq \tau_\emptyset$) in the target tree, and vice versa.*

5. *Monotonous: For $\langle \tau_i^s, \tau_i^t\rangle, \langle \tau_j^s, \tau_j^t\rangle \in \mathbb{H}_L$, $\tau_i^s \in ds(\tau_j^s)$ iff $\tau_i^t \in ds(\tau_j^t)$.*

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

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 $\mathbb{h}_m, \mathbb{h}_n \in \mathbb{H}_L$ are in positions of a parse tree that can support $\mathbb{h}_i$, $\mathbb{h}_i$ and the support relation should be added to $\langle \mathbb{H}_L, \Delta_L\rangle$. 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 Δ_L to be further restricted so that each of aligned pairs in $\mathbb{H}_L$ has only one support.

Let us assume that alignment $\mathbb{h}_i$ is supported by more than one pair of descendant alignments

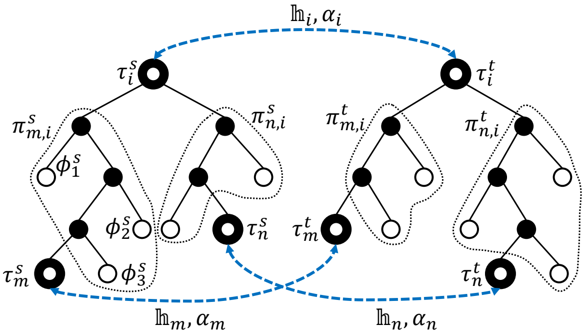

Figure 3: Inside probability depends on support alignments and paths to reach an LCA.

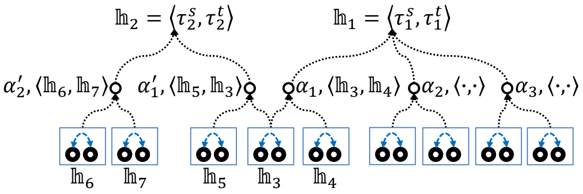
$\mathbb{h}_2 = \langle\tau_2^s, \tau_2^t\rangle$
$\mathbb{h}_1 = \langle\tau_1^s, \tau_1^t\rangle$
$\alpha'_2, \langle\mathbb{h}_6, \mathbb{h}_7\rangle$
$\alpha'_1, \langle\mathbb{h}_5, \mathbb{h}_3\rangle$
$\alpha_1, \langle\mathbb{h}_3, \mathbb{h}_4\rangle$
$\alpha_2, \langle\cdot,\cdot\rangle$
$\alpha_3, \langle\cdot,\cdot\rangle$
$\mathbb{h}_6$
$\mathbb{h}_7$
$\mathbb{h}_5$
$\mathbb{h}_3$
$\mathbb{h}_4$

Figure 4: Alignment pairs and packed supports

in Δ_L, *i.e.*, $\Delta_L \supseteq (\{\langle\mathbb{h}_m, \mathbb{h}_n\rangle\} \Rightarrow \mathbb{h}_i)$[1]. We denote $\mathbb{H}_m = \{\mathbb{h}_m\}$ and $\mathbb{H}_n = \{\mathbb{h}_n\}$. For each $\mathbb{h}_m \in \mathbb{H}_m$ and $\mathbb{h}_n \in \mathbb{H}_n$, we remove all support relations from Δ_L except for the maximum pairs or the pre-terminal alignments. The resultant set Δ'_L satisfies:

Theorem 3.2. *For all* $(\langle\mathbb{h}_m, \mathbb{h}_n\rangle \Rightarrow \mathbb{h}_i) \in \Delta'_L$, $\tau_i^s = lca(\tau_m^s, \tau_n^s)$ *and* $\tau_i^t = lca(\tau_m^t, \tau_n^t)$ *are true.*

In Fig. 2, τ_i^s is the lowest common ancestor (LCA) of τ_m^s and τ_n^s, and τ_i^t is the LCA of τ_m^t and τ_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 $\mathbb{h}_i$ is aligned from descendant alignments, *i.e.*, $\mathbb{h}_m$ and $\mathbb{h}_n$.

4.1 Probabilistic Model

Similar to the probabilistic context free grammar (PCFG), the inside probability α_i of $\mathbb{h}_i$ is determined by the inside probabilities, α_m and α_n, of the support pairs, together with the probability of the rule, *i.e.*, the way by which $\mathbb{h}_m$ and $\mathbb{h}_n$ are combined to support $\mathbb{h}_i$ as shown in Fig. 3. It is characterized by four paths, $\pi^s_{m,i}$ (the path from τ_m^s to τ_i^s), $\pi^s_{n,i}$ (τ_n^s to τ_i^s), $\pi^t_{m,i}$ (τ_m^t to τ_i^t), and $\pi^t_{n,i}$ (τ_n^t to τ_i^t).

Each path consists of a set of null-aligned phrases $\phi \in \langle\phi, \tau_\emptyset\rangle$ and their mothers, *e.g.*, the path $\pi^s_{m,i}$ in Fig. 3 is a set of $\langle\phi_1^s, m(\phi_1^s)\rangle$, $\langle\phi_2^s, m(\phi_2^s)\rangle$, and $\langle\phi_3^s, m(\phi_3^s)\rangle$. We assume that each occurrence of a null-alignment is independent. Thus, its probability $\beta^s_{m,i}$ is computed as:

$$\beta^s_{m,i} = \Pi_{\phi_k^s \in \pi^s_{m,i}} P_r(\phi_k^s, \tau_\emptyset).$$

$\beta^s_{n,i}$, $\beta^t_{m,i}$, and $\beta^t_{n,i}$ are computed in the same manner. We abbreviate $\gamma^s_{m,n,i} = \beta^s_{m,i}\beta^s_{n,i}$, likewise $\gamma^t_{m,n,i} = \beta^t_{m,i}\beta^t_{n,i}$. Finally, α_i can be represented as a simple relation:

$$\alpha_i = \alpha_m \alpha_n P_r(\tau_i^s, \tau_i^t)\gamma^s_{m,n,i}\gamma^t_{m,n,i}. \qquad (1)$$

$P_r(\cdot,\cdot)$ is the alignment probability parameterized in Sec. 5. Since we assume that the structures of parse trees of s and t are determined by a parser, the values of $\gamma^s_{m,n,i}$ and $\gamma^t_{m,n,i}$ are fixed. Therefore, by traversing the parse tree in a bottom-up manner, we can identify an LCA (*i.e.*, τ_i) for phrases τ_m and τ_n while simultaneously computing $\gamma_{m,n,i}$.

4.2 Alignment Algorithm

Algorithm 4.1 depicts our algorithm. Given word alignments $\mathbb{W} = \{\langle w_i^s, w_i^t\rangle\}$, it constructs legitimate sets of aligned pairs in a bottom-up manner. Like the CKY algorithm, Algorithm 4.1 uses DP to efficiently compute all possible legitimate sets and their probabilities in parallel. In addition, null-alignments are allowed when aligning an LCA supported by aligned descendant nodes.

$A[\cdot]$ is indexed by phrases in the parse tree of s and maintains a list of all possible aligned pairs. Furthermore, to deal with non-monotonic alignment (Sec. 4.3), it keeps all competing hypotheses of support relations using packed representations. Specifically, $\mathbb{h}_i$ is accompanied by its packed support list as illustrated in Fig. 4; $\mathbb{h}_1 = \langle\tau_1^s, \tau_1^t\rangle$ is aligned with supports of $\{\langle\alpha_j, \langle\mathbb{h}_m, \mathbb{h}_n\rangle\rangle\}$ like $\langle\alpha_1, \langle\mathbb{h}_3, \mathbb{h}_4\rangle\rangle$. Depending on the support alignments, $\mathbb{h}_i$ has different inside probabilities, *i.e.*, α_1, α_2, and α_3. Since the succeeding process of alignment only deals with the LCA's of τ_1^s and τ_1^t that are independent of the support alignment, all

[1] $\Rightarrow$ and $\overset{R}{\Rightarrow}$ are not distinguished here.

Algorithm 4.1 Phrase Alignment

```
LCAs and $\gamma$ in parse trees of $s$ and $t$ are computed and stored in $Lca^s[\cdot][\cdot]$ and $Lca^t[\cdot][\cdot]$.
set $A[\tau^s] \leftarrow \emptyset$ for all $\tau^s$
for all $\langle w^s, w^t \rangle \in \mathbb{W}$ do
    Find $\tau^s$ and $\tau^t$ covering $w^s$ and $w^t$
    Compute $\alpha_i$ of $\langle \tau^s, \tau^t \rangle$ using Eq. (1)
    PACK($\langle \tau^s, \tau^t \rangle, \langle \alpha_i, \emptyset \rangle, A$)
for all $\tau^s_m, \tau^s_n$ do ▷ Trace the source tree from the bottom to top
    for all $\langle \tau^s_i, \gamma^s_{m,n,i} \rangle \in Lca^s[\tau^s_m][\tau^s_n]$ do
        ALIGN($\tau^s_m, \tau^s_n, \tau^s_i, \gamma^s_{m,n,i}, A$)
function ALIGN($\tau^s_m, \tau^s_n, \tau^s_i, \gamma^s, A$)
    for all $\mathbb{h}_m = \langle \tau^s_m, \tau^t_m \rangle \in A[\tau^s_m]$ do
        for all $\mathbb{h}_n = \langle \tau^s_n, \tau^t_n \rangle \in A[\tau^s_n]$ do
            $\langle \tau^t_i, \gamma^t \rangle \leftarrow Lca^t[\tau^t_m][\tau^t_n]$
            Compute $\alpha_i$ using Eq. (1)
            PACK($\langle \tau^s_i, \tau^t_i \rangle, \langle \alpha_i, \langle \mathbb{h}_m, \mathbb{h}_n \rangle \rangle, A$)
function PACK($\langle \tau^s, \tau^t \rangle, \langle \alpha, \langle \mathbb{h}_m, \mathbb{h}_n \rangle \rangle, A$)
    if $\langle \tau^s, \tau^t \rangle \in A[\tau^s]$ then
        $A[\tau^s] \leftarrow A[\tau^s] \cup \langle \alpha, \langle \mathbb{h}_m, \mathbb{h}_n \rangle \rangle$ ▷ Merge supports and their inside probability
    else
        $A[\tau^s] \leftarrow (\langle \tau^s, \tau^t \rangle, \langle \alpha, \langle \mathbb{h}_m, \mathbb{h}_n \rangle \rangle)$
```

support relations are packed as a support list[2] by the PACK function.

4.3 Non-Compositional Alignment

A monotonic alignment requires $\tau^t_m \in \mathbb{h}_m$ and $\tau^t_n \in \mathbb{h}_n$ 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.*, τ^t_m) is an ancestor of another (*e.g.*, τ^t_n) or the same phrase. Such a case could be exceptionally compatible, when τ^t_m has null-alignments and all the aligned phrases of τ^t_n fit in these null-alignments. A new alignment $\langle \tau^s_i, \tau^t_i (= \tau^t_m) \rangle$ would be non-monotonically formed. Fig. 5 shows a real example of non-compositional alignment produced by our method. The target phrase τ^t_n ("through the spirit of teamwork") is null-alignment when aligning τ^s_m and τ^t_m, but then the alignment to τ^s_n ("Relying on team spirit") is allowed by non-compositional alignment of τ^s_i.

Algorithm 4.2 Non-Compositional Alignment

```
1:  function TRACE($\tau_n, \tau_m$)    ▷ $\tau_n \in ds(\tau_m)$
2:      $V \leftarrow \emptyset$
3:      for all $[\tau_m]^i$ do
4:          if $\tau_n \in ds(\phi)$ for $\exists \phi \in \Phi^{[\tau_m]^i}$ then
5:              $V \leftarrow V \cup \langle \Psi^{[\tau_m]^i} \cup \tau_n, (\Phi^{[\tau_m]^i} \setminus \phi) \cup$ GAP$(\tau_n, \phi)$ $\rangle$
6:          else if $\tau_n \in ds(\psi)$ for $\exists \psi \in \Psi^{[\tau_m]^i}$ then
7:              $V \leftarrow V \cup$ TRACE$(\tau_n, \psi)$
8:          else
9:              for all $[\tau_n]^j$ do
10:                 $V \leftarrow V \cup$ DOWN$([\tau_n]^j, [\tau_m]^i)$
11:     return $V$;
```

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

Algorithm 4.2 checks if target phrases τ_m and $\tau_n \in ds(\tau_m)$ are compatible. We use the following notations: $[\tau_m]^i$ and $[\tau_n]^j$ represent the phrases of τ_m and τ_n with the i-th and j-th sets of supporting alignments, respectively. For τ^t_2 in Fig. 4, there are $[\tau^t_2]^1$ supported by $\langle \mathbb{h}_5, \mathbb{h}_3 \rangle$ and $[\tau^t_2]^2$ supported by $\langle \mathbb{h}_6, \mathbb{h}_7 \rangle$. $[\tau_m]^i$ consists of sets of aligned target phrases $\Psi^{[\tau_m]^i} = \{\psi_k^{[\tau_m]^i}\}$ and null-alignments $\Phi^{[\tau_m]^i} = \{\phi_l^{[\tau_m]^i}\}$ ($[\tau_n]^j$ is similar).

For each $[\tau_m]^i$, if τ_n 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 τ_n is a descendant of a support of τ_m, the compatibility is recursively checked (line 7). Otherwise, the compatibility of the supports of τ_n and τ_m are recursively checked in DOWN function in a similar manner (line 10).

When TRACE function returns a set of $\{\langle \Psi^k, \Phi^k \rangle\}$, 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.

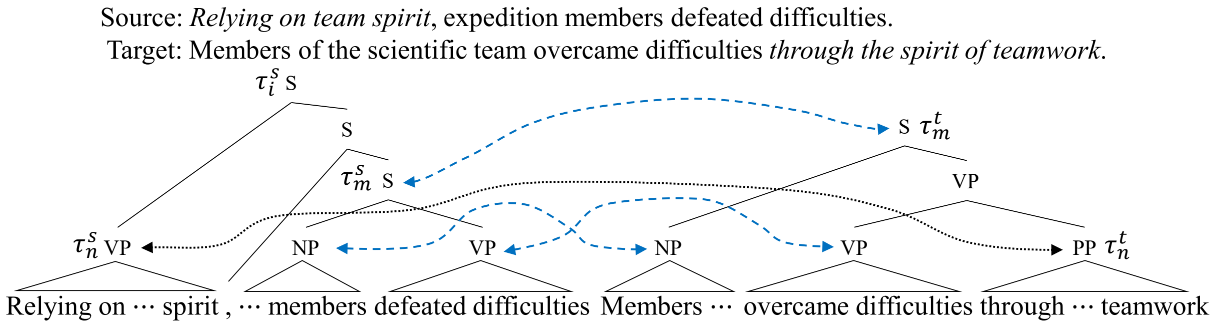
Source: *Relying on team spirit*, expedition members defeated difficulties.
Target: Members of the scientific team overcame difficulties *through the spirit of teamwork*.
τ_i^s S
S
τ_m^s S
S τ_m^t
VP
τ_n^s VP
NP
VP
NP
VP
PP τ_n^t
Relying on ⋯ spirit , ⋯ members defeated difficulties Members ⋯ overcame difficulties through ⋯ teamwork

Figure 5: Example of a non-compositional alignment

gether with a new alignment pair $\langle\tau_i^s,\tau_i^t\rangle$ where $\tau_i^s = lca(\tau_m^s,\tau_n^s)$ and $\tau_i^t = \tau_m^t$.

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 reranking 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:

$$P_r(\tau_i^s,\tau_i^t) \doteq \frac{\exp(\mathbf{w}\cdot\mathbb{F}(\boldsymbol{a}_i^s,\boldsymbol{a}_i^t))}{\sum_{\langle\tau_j^s,\tau_j^t\rangle,\tau_i^s=\tau_j^s}\exp(\mathbf{w}\cdot\mathbb{F}(\boldsymbol{a}_j^s,\boldsymbol{a}_j^t))},$$

where $\boldsymbol{a} \doteq (a_0,\cdots,a_n)$ consists of n attributes of τ. $\mathbb{F}(\cdot,\cdot)$ and $\mathbf{w}$ 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 $a_{\text{lex}} \in \boldsymbol{a}$ combined with its syntactic category $a_{\text{cat}} \in \boldsymbol{a}$ is encoded as a feature[3] as shown below. We use semantic (instead of syntactic) heads to encode semantic relationships in paraphrases.

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

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[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 at each category pair, while the seventh feature 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 $\tau_{\emptyset}$), 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:

$$P_e(\tau_i^s, \tau_i^t) = \begin{cases} \exp\{-(|\tau_i^s|_\phi + |\tau_i^t|_\phi + \mu_c + 1)^{\mu_n}\} \\ \qquad \text{(non-compositional alignment)} \\ \exp\{-(|\tau_i^s|_\phi + |\tau_i^t|_\phi + 1)^{\mu_n}\} \\ \qquad \text{(otherwise)} \end{cases}$$

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 $\langle \tau_i^s, \tau_i^t \rangle$, the overall probability is computed as:

$$(1 - \mu_p)\alpha_i + \mu_p \varrho(\tau_i^s)\varrho(\tau_i^t),$$

where $\varrho(\cdot)$ 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[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 $\mathbb{W}$ 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 $\mathbb{W}$ using word embeddings; we use the MVLSA 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

[4] http://wordnet.princeton.edu

[5] NIST OpenMT corpora: LDC2010T14, LDC2010T17, LDC2010T21, LDC2010T23, LDC2013T03

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:

$$\text{Recall} = \frac{|\{\mathbb{h}|\mathbb{h} \in \mathbb{H}_a \wedge \mathbb{h} \in \mathbb{G} \cap \mathbb{G}'\}|}{|\mathbb{G} \cap \mathbb{G}'|},$$

$$\text{Precision} = \frac{|\{\mathbb{h}|\mathbb{h} \in \mathbb{H}a \wedge \mathbb{h} \in \mathbb{G} \cup \mathbb{G}'\}|}{|\mathbb{H}a|},$$

where $\mathbb{H}a$ is a set of alignments, while $\mathbb{G}$ and $\mathbb{G}'$ are the ones that two of annotators produce, respectively. The function of $|\cdot|$ counts the elements in a set. There are three combinations for $\mathbb{G}$ and $\mathbb{G}'$ 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.

	Roles of hyper-parameters
μ_n	Control penalty for null-alignment
μ_c	Control penalty for non-compositional alignment
μ_p	Balance alignment and parsing prob.
μ_b	Beam size at alignment
μ_g	Generation gap to reach an LCA

Table 2: Summary of the hyper-parameters

Method	Recall	Prec.	UAS	%
Human	90.65	88.21	–	–
Proposed	**83.64**	**78.91**	93.49	98
Monotonic	82.86*	77.97*	93.49	98
w/o EM	81.33*	75.09*	92.91*	86
1-best tree	80.11*	73.26*	93.56	100

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-

[6] http://ilk.uvt.nl/conll/software.html

[7] Although omitted, the labeled attachment score showed the same tendency as UAS.

annotator agreement, since the conventional inter-annotator 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 against the model trained on $2K$ samples. These results demonstrate that our method can be trained with a moderate amount of data.

Observations Previous studies show that parallel parsing improves parsing quality, while such an effect is insignificant here. We examine causes through manual observations.

The evaluation script indicated that our method corrected 34 errors while introducing 41 new errors[8]. We further analyzed these 75 cases; 12 cases are ambiguous as both the gold-standard and the output are correct. In addition, 8 cases are due to erroneous original sentences that should be disregarded, *e.g.*, " For two weeks ago,..." and "According 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

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 582–590, Los Angeles, California.

Or Biran, Terra Blevins, and Kathleen McKeown. 2016. Mining paraphrasal typed templates from a plain text corpus. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1913–1923, Berlin, Germany.

Peter F. Brown, Stephen 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.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceesings of the Conference on Natural Language Learning (CoNLL)*, pages 149–164, New York City.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 127–135, Los Angeles, California.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceesings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 877–886, Honolulu, Hawaii.

Gideon Maillette de Buy Wenniger and Khalil Sima'an. 2013. A formal characterization of parsing word alignments by synchronous grammars with empirical evidence to the ITG hypothesis. In *Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, pages 58–67, Atlanta, Georgia.

Do Kook Choe and David McClosky. 2015. Parsing paraphrases with joint inference. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1223–1233, Beijing, China.

John Cocke. 1969. *Programming Languages and Their Compilers: Preliminary Notes*. Courant Institute of Mathematical Sciences, New York University.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 468–476, Suntec, Singapore.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceesings of the International Conference on Computational Linguistics (COLING)*, pages 350–356, Geneva, Switzerland.

Manjuan Duan, Ethan Hill, and Michael White. 2016. Generating disambiguating paraphrases for structurally ambiguous sentences. In *Proceedings of the Linguistic Annotation Workshop (LAW)*, pages 160–170, Berlin, Germany.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 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, Atlanta, Georgia.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1011–1019, Los Angeles, California.

Miriam Kaeshammer. 2013. Synchronous linear context-free rewriting systems for machine translation. In *Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, pages 68–77, Atlanta, Georgia.

Tadao Kasami. 1965. An efficient recognition and syntax-analysis algorithm for context-free languages. Scientific report AFCRL-65-758, Air Force Cambridge Research Lab.

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 611–619, Boulder, Colorado.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 802–811, Honolulu, Hawaii.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 556–566, Denver, Colorado.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan.

Kenji Sagae, Yusuke Miyao, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Challenges in mapping of syntactic representations for framework-independent parser evaluation. In *Proceedings of the Workshop on Automated Syntatic Annotations for Interoperable Language Resources*.

David Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation (WMT)*, pages 23–30, New York City.

David A. Smith and Noah A. Smith. 2004. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 49–56, Barcelona, Spain.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, Seattle, Washington, USA.

Vivek Srikumar and Dan Roth. 2013. Modeling semantic relations expressed by prepositions. *Transactions of the Association of Computational Linguistics (TACL)*, 1:231–242.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. pages 1556–1566, Beijing, China.

Kapil Thadani, Scott Martin, and Michael White. 2012. A joint phrasal and dependency model for paraphrase alignment. In *Proceesings of the International Conference on Computational Linguistics (COLING)*, pages 1229–1238, Mumbai, India.

Jonathan Weese, Juri Ganitkevitch, and Chris Callison-Burch. 2014. PARADIGM: Paraphrase diagnostics through grammar matching. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 192–201, Gothenburg, Sweden.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association of Computational Linguistics (TACL)*, 3(1):345–358.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceesings of the International Conference on Learning Representations (ICLR)*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Semi-Markov phrase-based monolingual alignment. In *Proceesings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 590–600, Seattle, Washington, USA.

Wenpeng Yin and Hinrich Schütze. 2015. Discriminative phrase embedding for paraphrase identification. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1368–1373, Denver, Colorado.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.