

Acquisition of Phrase Correspondences using Natural Deduction Proofs

Hitomi Yanaka¹

hitomiyataka@g.ecc.u-tokyo.ac.jp

Koji Mineshima²

mineshima.koji@ocha.ac.jp

Pascual Martínez-Gómez³

pascual.mg@aist.go.jp

Daisuke Bekki²

bekki@is.ocha.ac.jp

¹The University of Tokyo

²Ochanomizu University

³Artificial Intelligence Research Center, AIST
Tokyo, Japan

Abstract

How to identify, extract, and use phrasal knowledge is a crucial problem for the task of Recognizing Textual Entailment (RTE). To solve this problem, we propose a method for detecting paraphrases via natural deduction proofs of semantic relations between sentence pairs. Our solution relies on a graph reformulation of partial variable unifications and an algorithm that induces subgraph alignments between meaning representations. Experiments show that our method can automatically detect various paraphrases that are absent from existing paraphrase databases. In addition, the detection of paraphrases using proof information improves the accuracy of RTE tasks.

1 Introduction

Recognizing Textual Entailment (RTE) is a challenging natural language processing task that aims to judge whether one text fragment logically follows from another text fragment (Dagan et al., 2013). Logic-based approaches have been successful in representing the meanings of complex sentences, ultimately having a positive impact on RTE (Bjerva et al., 2014; Beltagy et al., 2014; Mineshima et al., 2015, 2016; Abzianidze, 2015, 2016). Although logic-based approaches succeed in capturing the meanings of functional or logical words, it is difficult to capture the meanings of content words or phrases using genuine logical inference alone. This remains a crucial problem in accounting for lexical relations between content words or phrases via logical inference. To solve this problem, previous logic-based approaches use knowledge databases such as WordNet (Miller, 1995) to identify lexical relations within a sen-

tence pair. While this solution has been successful in handling word-level paraphrases, its extension to phrase-level semantic relations is still an unsolved problem. There are three main difficulties that prevent an effective identification and use of phrasal linguistic knowledge.

The first difficulty is the presence of out-of-context phrase relations in popular databases such as the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013). PPDB may suggest paraphrases that do not adhere to the context of our relevant text segments nor to their semantic structure, which might be problematic.

The second difficulty is finding semantic phrase correspondences between the relevant text segments. Typical approaches only rely on surface (Beltagy et al., 2013) or syntactic correspondences (Arase and Tsujii, 2017), often producing inaccurate alignments that significantly impact our inference capabilities. Instead, a mechanism to compute semantic phrase correspondences could potentially produce, if available, more coherent phrase pairs and solve the recurring issue of discontinuity.

The third difficulty is the intrinsic lack of coverage of databases for logical inference despite their large size. Whereas there is a relatively small number of possible word-to-word correspondences and thus their semantic relations can be enumerated, the same is not true for all phrase pairs that might be of interest. One alternative is to use functions of infinite domain (e.g., cosine similarity) between phrase representations (Tian et al., 2016), but these techniques are still under development, and we have not seen definitive successful applications when combined with logic systems.

In this study, we tackle these three problems. The contributions of this paper are summarized as follows: First, we propose a new method of detecting phrase correspondences through natu-

ral deduction proofs of semantic relations for a given sentence pair. Second, we show that our method automatically extracts various paraphrases that compensate for a shortage in previous paraphrase databases. Experiments show that extracted paraphrases using proof information improve the accuracy of RTE tasks.

2 Related Work

In this section, we review previous logical inference systems that are combined with lexical knowledge. The RTE system developed by Abzianidze (2016) uses WordNet as axioms and adds missing knowledge manually from the training dataset; however, this technique requires considerable human effort and is not extended to handle phrasal knowledge.

Martínez-Gómez et al. (2017) proposed an RTE system with an on-the-fly axiom injection mechanism guided by a natural deduction theorem prover. Pairs of unprovable sub-goals and plausible single premises are identified by means of a variable unification routine and then linguistic relations between their logical predicates are checked using lexical knowledge such as WordNet and VerbOcean (Chklovski and Pantel, 2004). However, this mechanism is limited to capturing word-to-word relations within a sentence pair.

Bjerva et al. (2014) proposes an RTE system where WordNet relations are used as axioms for word-to-word knowledge in theorem proving. For phrasal knowledge, PPDB is used to rephrase an input sentence pair instead of translating paraphrases into axioms. However, this solution ignores logical contexts that might be necessary when applying phrasal knowledge. Moreover, it does not apply to discontinuous phrases.

Beltagy et al. (2016) uses WordNet and PPDB as lexical knowledge in the RTE system. To increase their coverage of phrasal knowledge, the system combines a resolution strategy to align clauses and literals in a sentence pair and a statistical classifier to identify their semantic relation. However, this strategy only considers one possible set of alignments between fragments of a sentence pair, possibly causing inaccuracies when there are repetitions of content words and meta-predicates.

In our research, we propose an automatic phrase abduction mechanism to inject phrasal knowledge during the proof construction process. In addition, we consider multiple alignments by backtracking

the decisions on variable and predicate unifications, which is a more flexible strategy. We represent logical formulas using graphs, since this is a general formalism that is easy to visualize and analyze. However, we use *natural deduction* (see Section 3.2) as a proof system instead of Markov Logic Networks for inference. Some research has investigated graph operations for semantic parsing (Reddy et al., 2014, 2016) and abstractive summarization (Liu et al., 2015); we contribute to these ideas by proposing a subgraph mapping algorithm that is useful for performing natural language inferences.

Considerable research efforts have been focused on the identification and extraction of paraphrases. One successful technique is associated with bilingual pivoting (Bannard and Callison-Burch, 2005; Zhao et al., 2008), in which alternative phrase translations are used as paraphrases at a certain probability. However, this technique requires large bilingual parallel corpora; moreover, word alignment errors likely cause noisy paraphrases. Another strategy is to extract phrase pairs from a monolingual paraphrase corpus using alignments between syntactic trees, guided by a linguistically motivated grammar (Arase and Tsujii, 2017). The main difference between these studies and ours is that they typically attempt alignment between words or syntactic trees, whereas we perform alignments between meaning representations, which enables the acquisition of more general paraphrases by distinguishing functional words from content words. This point is important in distinguishing among different semantic relations (e.g., antonyms and synonyms). In addition, word and syntactic alignments potentially ignore coreferences, making it difficult to find relations between many-to-many sentences. Semantic alignments enable this because coreferences must refer to the same variable as the original entity.

3 Logic-based Approach to RTE

3.1 Meaning representation

In logic-based approaches to RTE, a text T and a hypothesis H are mapped onto logical formulas T' and H' . To judge whether T entails H , we check whether $T' \Rightarrow H'$ is a theorem in a logical system.

For meaning representations, we use Neo-Davidsonian event semantics (Parsons, 1990). In this approach, a verb is analyzed as a one-place predicate over events. Both the arguments of a

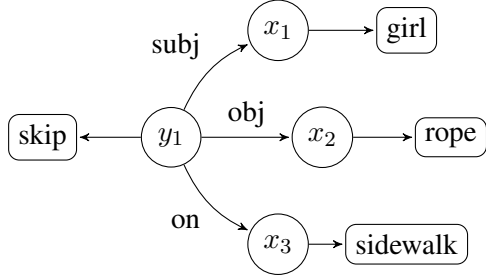


Figure 1: A graph for the basic formula (2).

verb and modifiers are linked to events by semantic roles, and the entire sentence is closed by existential quantification over events. For example, (1) is mapped onto (2).

- (1) A girl is skipping rope on a sidewalk.
- (2) $\exists x_1 \exists x_2 \exists x_3 \exists y_1 (\text{girl}(x_1) \wedge \text{rope}(x_2) \wedge \text{sidewalk}(x_3) \wedge \text{skip}(y_1) \wedge (\text{subj}(y_1) = x_1) \wedge (\text{obj}(y_1) = x_2) \wedge \text{on}(y_1, x_3))$

We use x_i as a variable for entities and y_j for events. In this semantics, we represent all content words (e.g., *girl* and *skip*) as one-place predicates. Regarding functional words, we represent a preposition like *on* as a two-place predicate, e.g., $\text{on}(y_1, x_3)$. We also use a small set of semantic roles such as **subj** and **obj** as a functional term and use equality ($=$) to connect an event and its participant, as in $\text{subj}(y_1) = x_1$.

To be precise, the set of *atomic formulas* \mathcal{A} in this event semantics is defined by the rule

$$\mathcal{A} ::= \mathbf{F}(t) \mid \mathbf{G}(t, u) \mid t = u$$

where $\mathbf{F}(t)$ is a one-place predicate (for content words), $\mathbf{G}(t, u)$ is a two-place predicate (for prepositions), t and u are a term. A term is defined as a constant, a variable, or a functional term of the form $f(t)$ where f is a semantic role and t is a term.

We call a formula constructed by conjunctions and existential quantifiers a *basic formula* in event semantics. Thus, a set of basic formulas φ in event semantics is defined as:

$$\varphi ::= \mathcal{A} \mid \varphi \wedge \varphi \mid \exists t \varphi$$

The formula in (2) is an instance of a basic formula, which captures the predicate-argument structure of a sentence.

On top of the system of basic formulas, we have a full language of event semantics with negation (\neg), disjunction (\vee), implication (\rightarrow), and a universal quantifier (\forall). These operators are used to represent additional logical features.

There is a natural correspondence between basic formulas and directed acyclic graphs (DAGs). Figure 1 shows an example¹. In the graph representation, constants and variables correspond to vertices; both two-place predicates for prepositions (e.g., $\text{on}(y_1, x_1)$) and functional terms for semantic roles (e.g., $\text{subj}(y_1) = x_1$) are represented as edges. A one-place predicate $\mathbf{F}(t)$ in a logical formula can be represented as a functional relation $\text{isa}(t, \mathbf{F})$, where isa is an expression relating a term t and a predicate \mathbf{F} represented as a vertex. The isa edges are unlabeled for simplicity.

3.2 Natural deduction and word abduction

We use the system of natural deduction (Prawitz, 1965; Troelstra and Schwichtenberg, 2000) to capture phrase correspondences from a sentence pair (T, H) , following the strategies for word axiom injection developed by Martínez-Gómez et al. (2017) and Yanaka et al. (2017). The sentence pair (T, H) is first mapped to a pair of formulas (T', H') . T' is initially set to the premise P , and H' is set to the goal G to be proved.

If formulas P and G are basic formulas, then the proving strategy is to decompose them into a set of atomic formulas using inference rules for conjunctions and existential quantifiers. The premise P is decomposed into a pool of premises $\mathcal{P} = \{\mathbf{p}_i(\theta_i) \mid i \in \{1, \dots, m\}\}$, where each $\mathbf{p}_i(\theta_i)$ is an atomic formula and θ_i is a list of terms appearing in $\mathbf{p}_i(\theta_i)$. The goal G is also decomposed into a set of sub-goals $\mathcal{G} = \{\mathbf{g}_j(\theta'_j) \mid j \in \{1, \dots, n\}\}$, where θ'_j is a list of terms appearing in $\mathbf{g}_j(\theta'_j)$.

The proof is performed by searching for a premise $\mathbf{p}_i(\theta_i)$ whose predicate matches that of a sub-goal $\mathbf{g}_j(\theta'_j)$. If such a premise is found, then variables in θ'_j are unified to those in θ_i and the sub-goal $\mathbf{g}_j(\theta'_j)$ can be removed from \mathcal{G} . If all the sub-goals can be removed, we prove $T' \rightarrow H'$. In the presence of two or more variables with the same predicate, there might be multiple possible variable unifications. Modern theorem provers explore these multiple possibilities in search of a configuration that proves a theorem.

Sub-goals may remain unproved when T logically does not entail H i.e., when there are no premise predicates \mathbf{p}_i that are matched with \mathbf{g}_j . In this case, the system tries word axiom injection, called *word abduction*. More specifically, if there

¹See Jones (2016) for some variants of graphical representations of logical formulas.

is a premise $\mathbf{p}_i(\theta_i)$ whose predicate has a linguistic relation (according to linguistic knowledge²) with that of a sub-goal $\mathbf{g}_j(\theta'_j)$, then variables in θ'_j are unified with those in θ_i and the sub-goal $\mathbf{g}_j(\theta'_j)$ can be removed from \mathcal{G} .

3.3 Graph illustration

Figure 2 shows an example to illustrate how the system works. To begin with, the input sentence pair (T, H) is mapped onto a pair of formulas, (T', H') . T' is initially placed to the premise P , and H' to the goal G . Note that these are basic formulas, and they are thus decomposed to the following sets of formulas \mathcal{P} and \mathcal{G} , respectively:

$$\begin{aligned} \mathcal{P} &= \{\mathbf{lady}(x_1), \mathbf{meat}(x_2), \mathbf{cut}(y_1), \mathbf{up}(y_1), \\ &\quad \mathbf{precisely}(y_1), \mathbf{subj}(y_1) = x_1, \mathbf{obj}(y_1) = x_2\} \\ \mathcal{G} &= \{\mathbf{woman}(x_3), \mathbf{meat}(x_4), \mathbf{cut}(y_2), \mathbf{piece}(x_5), \\ &\quad \mathbf{into}(y_2, x_5), \mathbf{subj}(y_2) = x_3, \mathbf{obj}(y_2) = x_4\} \end{aligned}$$

Steps 1 to 3 in Figure 2 demonstrate the variable unification routine and word axiom injection using graphs. Note that in step 1, all variables in formulas in \mathcal{P} or \mathcal{G} are initially different.

In step 2, we run a theorem proving mechanism that uses graph terminal vertices as anchors to unify variables between formulas in \mathcal{P} and those in \mathcal{G} . The premise $\mathbf{meat}(x_2)$ in \mathcal{P} matches the predicate \mathbf{meat} of the sub-goal $\mathbf{meat}(x_4)$ in \mathcal{G} and the variable unification $x_4 := x_2$ is applied (and similarly for the sub-goal $\mathbf{cut}(y_2)$ in \mathcal{G} with the variable unification $y_2 := y_1$).

In step 3, we use the previous variable unification on y_1 , the \mathbf{subj} edge in \mathcal{P} and \mathcal{G} and the axiom $\forall x. \mathbf{lady}(x) \rightarrow \mathbf{woman}(x)$ from external knowledge to infer that $x_3 := x_1$.

4 Phrase Abduction

There is one critical reason that the word-to-word axiom injection described in Section 3.2 fails to detect phrase-to-phrase correspondences. That is, the natural deduction mechanism decomposes the goal G into atomic sub-goals that are then proved *one-by-one* (word-by-word), independently of each other except for the variable unification effect. This mechanism is particularly problematic when we attempt to prove phrases that resist decomposition, two-place predicates (e.g., $\mathbf{into}(x, y)$), or failures in variable unification (e.g., due to inaccurate semantics). Thus, we propose a method to detect phrase-to-phrase correspondence through natural deduction proofs.

²As given in a linguistic ontology or database such as WordNet or VerbOcean.

4.1 Phrase pair detection

We detect phrase-to-phrase entailing relations between T' and H' by finding alignments between the subgraphs of their meaning representations when $T' \Rightarrow H'$ or $T' \Rightarrow \neg H'$ hold. Finding subgraph alignments is a generalization of the subgraph isomorphism problem, which is NP-complete³. In this paper, we approximate a solution to this problem by using a combination of a backtracking variable unification and a deterministic graph search on the neighborhood of non-unified variables.

Using our running example in Figure 2, step 4 displays our proposed subgraph alignment. The variable x_5 in the graph of \mathcal{G} cannot be unified with any variable in the graph of \mathcal{P} . This is a very common case in natural language inferences, as there might be concepts in H that are not directly supported by concepts in T . In this research, we propose spanning a subgraph starting at non-unified variables (e.g., x_5 in \mathcal{G}) whose boundaries are semantic roles (e.g., \mathbf{subj} , \mathbf{obj}). Its candidate semantics from \mathcal{P} are then the attributes of its corresponding unified variables from \mathcal{G} (e.g. $\mathbf{cut\ up\ precisely} \rightarrow \mathbf{cut\ into\ pieces}$).

4.2 Graph alignments

To formalize this solution we introduce some graph notation. Let $V = \mathcal{V}^u \cup \mathcal{V}^{\bar{u}} \cup \mathcal{L}$ be the set of vertices, where \mathcal{V}^u is the set of unified variables (e.g. x_1, x_2, y_1), $\mathcal{V}^{\bar{u}}$ is the set of non-unified variables (e.g. x_5), and \mathcal{L} is a set of predicates (e.g., \mathbf{lady} , \mathbf{woman}). Let E be the set of labeled, directed edges $\langle v, l, v' \rangle$ where $v, v' \in V$ and l are labels that may represent a functional relation \mathbf{isa} , a preposition or a semantic role. We denote a set of two-place predicates for prepositions as \mathbf{PREP} and a set of functional terms for semantic roles as \mathbf{ARGS} ; e.g., $\mathbf{ARGS} = \{\mathbf{subj}, \mathbf{obj}\}$. A graph that represents \mathcal{P} is then a tuple $G_{\mathcal{P}} = \langle V_{\mathcal{P}}, E_{\mathcal{P}} \rangle$, and similarly, for \mathcal{G} , $G_{\mathcal{G}} = \langle V_{\mathcal{G}}, E_{\mathcal{G}} \rangle$.

We can now define a function to span a subgraph in the neighborhood of non-unified variables $v \in \mathcal{V}^{\bar{u}}$ in the graph of \mathcal{G} . We call a connected set of edges in which no semantic roles appear, i.e., $\{\langle v, l, v' \rangle \mid l \notin \mathbf{ARGS}\}$, a *phrase set*. Let $E(x)$ be the phrase set in E such that each vertex is connected to x with an incoming or outgoing edge, that is, $E(x) = \{(v_i, l, v_k) \in E \mid (x = v_i \vee x = v_k) \wedge l \notin \mathbf{ARGS}\}$.

³Emmert-Streib et al. (2016) gives a good overview.

T : A lady is cutting up some meat precisely
 $T' : \exists x_1 \exists x_2 \exists y_1 (\text{lady}(x_1) \wedge \text{meat}(x_2) \wedge \text{cut}(y_1) \wedge \text{up}(y_1) \wedge \text{precisely}(y_1) \wedge \text{subj}(y_1, x_1) \wedge \text{obj}(y_1, x_2))$

H : Some meat is being cut into pieces by a woman
 $H' : \exists x_3 \exists x_4 \exists x_5 \exists y_2 (\text{meat}(x_4) \wedge \text{woman}(x_3) \wedge \text{cut}(y_2) \wedge \text{piece}(x_5) \wedge \text{into}(y_2, x_5) \wedge \text{subj}(y_2, x_3) \wedge \text{obj}(y_2, x_4))$

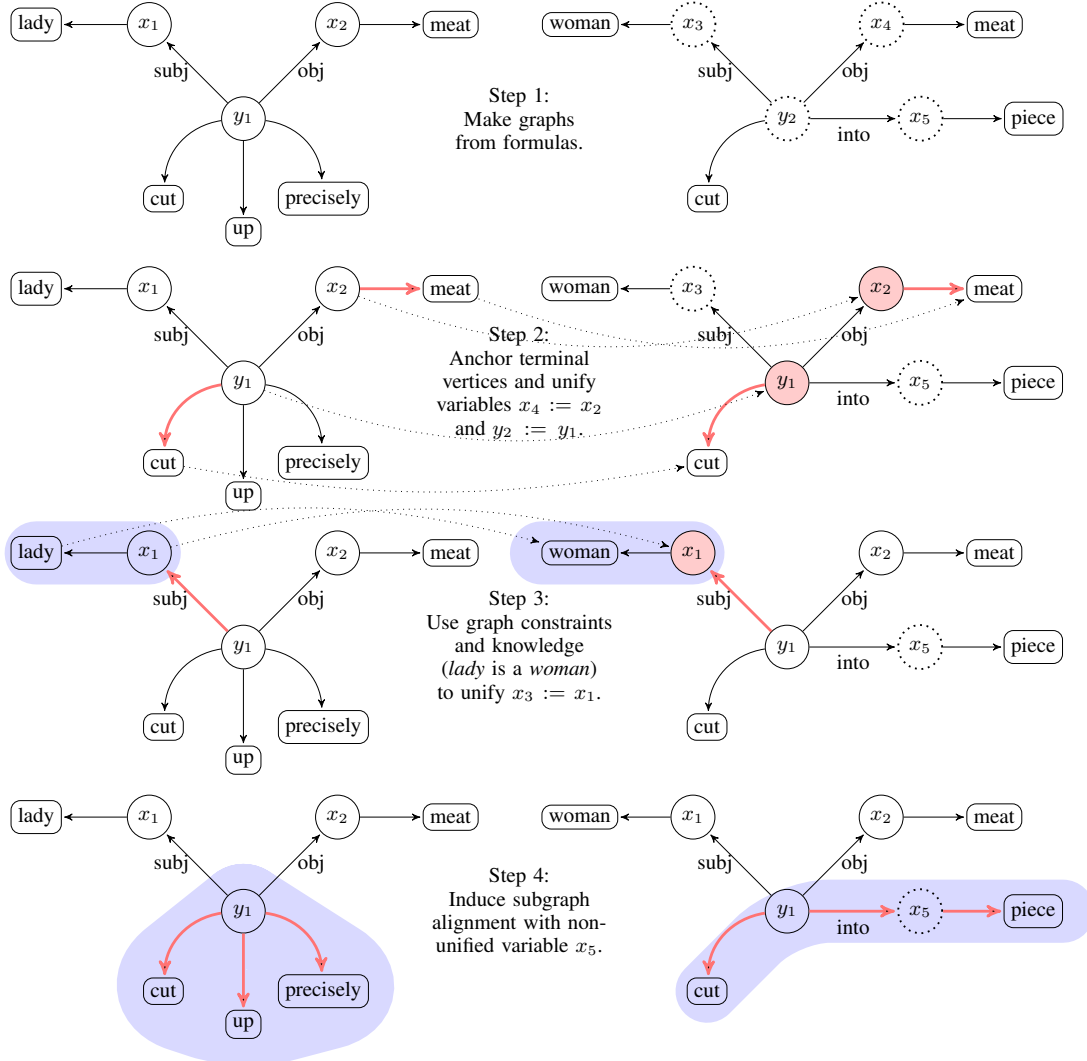


Figure 2: A graph representation of a theorem proving routine on basic formulas and variable unification. Dotted circles represent non-unified variables at each step, whereas edges without labels are attributes. The graph of the left side is the set of premises \mathcal{P} and the graph of the right side is the set of sub-goals \mathcal{G} . Colored subgraphs represent a word or a phrase to which our axiom injection mechanism applies.

Note that $E(x)$ induces a subgraph in a given graph G and the condition $l \notin \text{ARGS}$ sets the boundaries of the subgraph by excluding the semantic roles of verb phrases. Given two phrase sets E and E' , we say E' is reachable from E , written $E \sim E'$, if E and E' share at least one variable vertex. Let \sim^* be the transitive closure of \sim . Given a set of edges E_G and a variable v , we define the *extended phrase set*, written $\text{Reach}(v)$, as follows:

$$\text{Reach}(v) = \{e \in E \mid E_G(v) \sim^* E\}$$

that is, the set of edges e that can be reached from v without crossing an edge with a semantic role

label. This function defines a partition or equivalence class for non-unified variables $v \in \mathcal{V}_G^u$, and each of these partitions induce a (possibly discontinuous) phrase in \mathcal{G} that remains unproved.

The corresponding subgraph in \mathcal{P} to each of these partitions is given by the vertices and edges connected with a path of length one to the unified variables that appear in $\text{Reach}(v)$. That is,

$$\text{Corr}(v) = \{e \in E_{\mathcal{P}}(v'), v' \in V_G^{[v]} \cap V_{\mathcal{P}}\}$$

where $V_G^{[v]}$ denotes the vertices in the subgraph of \mathcal{G} induced by the partition $\text{Reach}(v)$.

A subgraph alignment between \mathcal{P} and \mathcal{G} is given by the pair of $\langle \text{Corr}(v), \text{Reach}(v) \rangle$ for all $v \in \mathcal{V}_{\mathcal{G}}^u$, where the phrases can be read from the predicates in the vertices and edges labeled with prepositions.

We define a mapping $(\cdot)^\bullet$ from a labeled edge $\langle v, l, v' \rangle$ to an atomic formula as follows.

$$\langle v, l, v' \rangle^\bullet = \begin{cases} v'(v) & \text{if } l \text{ is isa} \\ l(v, v') & \text{if } l \in \text{PREP} \\ l(v) = v' & \text{if } l \in \text{ARGS} \end{cases}$$

Let E be a set of labeled edges, and let E^\bullet be $\{\langle v, l, v' \rangle^\bullet \mid \langle v, l, v' \rangle \in E\}$. The phrase axiom generated for each non-unified variable $v \in \mathcal{V}_{\mathcal{G}}^u$ is defined as

$$\forall \theta_C. (\bigwedge \text{Corr}(v)^\bullet \rightarrow \exists \theta_R. (\bigwedge \text{Reach}(v)^\bullet)),$$

where θ_C is a set of free variables appearing in $\text{Corr}(v)^\bullet$ (which includes v) and θ_R is a set of free variables appearing in $\text{Reach}(v)^\bullet$ but not in $\text{Corr}(v)^\bullet$.

In Figure 2, the only non-unified variable in the sub-goal in step 4 is x_5 , that is, $\mathcal{V}_{\mathcal{G}}^u = \{x_5\}$. Then, starting from the variable x_5 , $\text{Reach}(x_5)$ is

$$\{\langle y_1, \text{into}, x_5 \rangle, \langle x_5, \text{isa}, \text{piece} \rangle\}.$$

Now $\mathcal{V}_{\mathcal{G}}^{[x_5]} = \{y_1, x_5\}$, and thus $\text{Corr}(x_5)$ is

$$\{\langle y_1, \text{isa}, \text{cut} \rangle, \langle y_1, \text{isa}, \text{up} \rangle, \langle y_1, \text{isa}, \text{precisely} \rangle\}.$$

Finally, the following is the axiom generated from $\langle \text{Corr}(x_5), \text{Reach}(x_5) \rangle^4$.

$$\forall y_1 (\text{cut}(y_1) \wedge \text{up}(y_1) \wedge \text{precisely}(y_1) \rightarrow \exists x_5 (\text{into}(y_1, x_5) \wedge \text{piece}(x_5))).$$

4.3 Non-basic formulas

If formulas P and G are not basic formulas (i.e., they contain logical operators other than \wedge and \exists), they are decomposed according to inference rules of natural deduction. There are two types of inference rules: introduction rules decompose a goal formula into smaller sub-goals, and elimination rules decompose a formula in the pool of premises into smaller ones. Figure 3 shows introduction rules and elimination rules for decomposing non-basic formulas including negation, disjunction, implication, and a universal quantifier. By applying inference rules, a proof of non-basic formulas appearing in sub-goals can be decomposed to a set of subproofs that only have basic formulas in sub-goals. If a universal quantifier appears in premises, it is treated in the same way as other premises.

⁴Note that this axiom is logically equivalent to

$$\forall y_1 (\text{cut}(y_1) \wedge \text{up}(y_1) \wedge \text{precisely}(y_1) \rightarrow \exists x_5 (\text{cut}(y_1) \wedge \text{into}(y_1, x_5) \wedge \text{piece}(x_5)))$$

indicated in the colored subgraphs in step 4 of Figure 2.

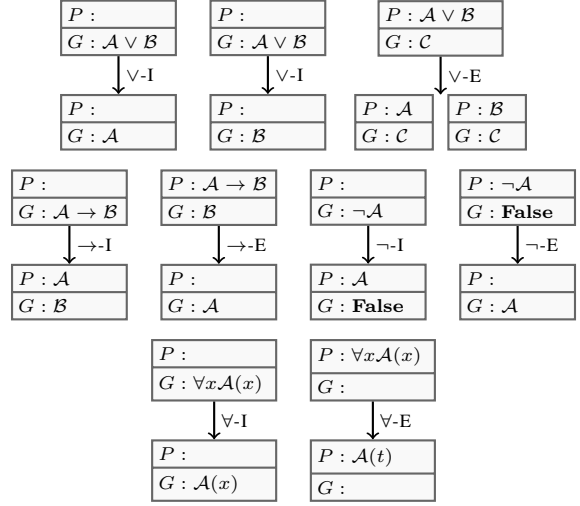


Figure 3: Inference rules used for decomposing non-basic formulas. P is a premise and G is a sub-goal. The initial formulas are at the top, with the formulas obtained by applying the inference rules shown below.

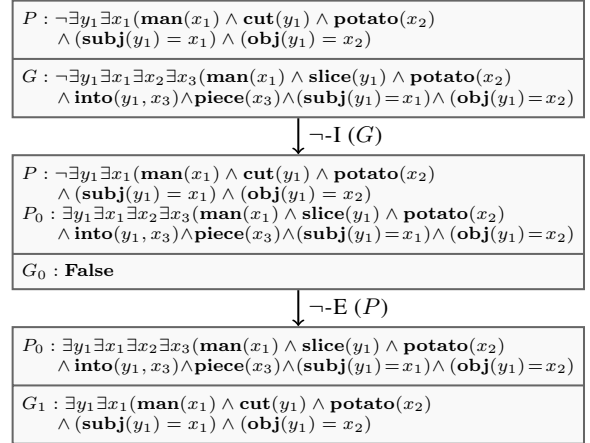


Figure 4: Proof process for the contradiction.

For example, consider the following sentence pair with the gold label “no” (contradiction):

T : A man is not cutting a potato

H : A man is slicing a potato into pieces

Figure 4 shows the proof process of $T' \Rightarrow \neg H'$. To prove the contradiction, the formulas T' and $\neg H'$ are set to P and G , respectively. Then, the negation in G is removed by applying the introduction rule (\neg -I) to G . Here, **False** is the propositional constant denoting the contradiction. In the second stage of the proof, the goal is to prove **False** in G_0 from the two premises P and P_0 . By applying (\neg -E) to P , we can eliminate the negation from P , resulting in the new goal G_1 .

As both the premise P_0 and the sub-goal G_1 are

basic formulas, the procedure described in the previous sections applies to the pair (P_0, G_1) ; these basic formulas are decomposed into atomic ones, and then the word-to-word abduction generates the desired axiom $\forall y_1(\text{cut}(y_1) \rightarrow \text{slice}(y_1))$. Finally, the graph alignment applies in the same way as described in Figure 2, which generates the phrase axiom:

$$\forall y_1(\text{cut}(y_1) \rightarrow \exists x_5(\text{into}(y_1, x_5) \wedge \text{piece}(x_5)))$$

Using this axiom, one can complete the proof of the contradiction between T' and H' .

5 Experiments

5.1 Dataset selection

We use the SemEval-2014 version of the SICK dataset (Marelli et al., 2014) for evaluation. The SICK dataset is a dataset for semantic textual similarity (STS) as well as for RTE. It was originally designed for evaluating compositional distributional semantics, so it contains logically challenging problems involving quantifiers, negation, conjunction, and disjunction, as well as inferences with lexical and phrasal knowledge.

The SNLI dataset (Bowman et al., 2015) contains inference problems requiring phrasal knowledge. However, it is not concerned with logically challenging expressions; the semantic relationships between a premise and a hypothesis are often limited to synonym/hyponym lexical substitution, replacements of short phrases, or exact word matching. This is because hypotheses are often parallel to the premise in structures and vocabularies. The FraCaS dataset (Cooper et al., 1994) also contains logically complex problems. However, it is confined to purely logical inferences and thus does not contain problems requiring inferences with lexical and phrasal knowledge. For these reasons, we choose the SICK dataset to evaluate our method of using logical inference to extract phrasal knowledge.

The SICK dataset contains 9927 sentence pairs with a 5000/4927 training/test split. These sentence pairs are manually annotated with three types of labels *yes* (entailment), *no* (contradiction), or *unknown* (neutral) (see Table 1 for examples). In RTE tasks, we need to consider a directional relation between words such as hypernym and hyponym to prove entailment and contradiction. Hence, to extract phrasal knowledge for RTE tasks, we use the training data whose gold label is

entailment or *contradiction*, excluding those with the *neutral* label.

5.2 Experimental setup

For the natural deduction proofs, we used *cgg2lambda* (Martínez-Gómez et al., 2016)⁵, a higher-order automatic inference system, which converts CCG derivation trees into semantic representations and conducts natural deduction proofs automatically. We parsed the tokenized sentences of the premises and hypotheses using three wide-coverage CCG parsers: C&C (Clark and Curran, 2007), EasyCCG (Lewis and Steedman, 2014), and *depccg* (Yoshikawa et al., 2017). CCG derivation trees (parses) were converted into logical semantic representations based on Neo-Davidsonian event semantics (Section 3.1). The validation of semantic templates used for semantic representations was conducted exclusively on the trial split of the SICK dataset. We used Coq (Bertot and Castran, 2010), an interactive natural deduction theorem prover that we run fully automatically with a number of built-in theorem-proving routines called tactics, which include first-order logic.

We compare phrase abduction with different experimental conditions. **No axioms** is our system without axiom injection. **W2W** is the previous strategy of word abduction (Martínez-Gómez et al., 2017). **P2P** is our strategy of phrase abduction; **W2W+P2P** combines phrase abduction with word abduction. In addition, we compare our system with three purely logic-based (unsupervised) approaches: **The Meaning Factory** (Bjerva et al., 2014), **LangPro** (Abzianidze, 2015), and **UTexas** (Beltagy et al., 2014). We also compare our system with machine learning-based approaches: the current state-of-the-art deep learning model **GRU** (Yin and Schütze, 2017), a log-linear regression model **SemEval-2014 best** (Lai and Hockenmaier, 2014), and a hybrid approach combining a logistic regression model and probabilistic logic **PL+eclassif** (Beltagy et al., 2016).

5.3 Extracted paraphrases

We extracted 9445 axioms from the SICK training dataset. The proving time average to extract phrasal axioms was only 3.0 seconds for a one-sentence pair⁶. Table 2 shows some examples of

⁵Available at <https://github.com/mynlp/cgg2lambda>.

⁶Ours is a polynomial-time instance of the graph matching problem, where the vertex cover set (maximum number of variables in a phrase) is bounded to a small constant.

ID	Text	Hypothesis	Entailment
3941	<i>A boy is looking at a calendar</i>	<i>There is nobody checking a calendar</i>	No
5938	<i>Vegetables are being put into a pot by a man</i>	<i>Someone is pouring ingredients into a pot</i>	Yes
5930	<i>The man is not doing exercises</i>	<i>Two men are fighting</i>	Unknown

Table 1: Examples in the SICK dataset with different entailment labels and similarity scores.

Kind	Text	Hypothesis
noun phrase	<i>A blond woman is sitting on the roof of a yellow vehicle, and two people are inside</i>	<i>A woman with blond hair is sitting on the roof of a yellow vehicle, and two people are inside</i>
verb phrase	<i>The person is setting fire to the cameras</i>	<i>Some cameras are being burned by a person with a blow torch</i>
verb phrase	<i>A man and a woman are walking together through the woods</i>	<i>A man and a woman are hiking through a wooded area</i>
prepositional phrase	<i>A child, who is small, is outdoors climbing steps outdoors in an area full of grass</i>	<i>A small child is outdoors climbing steps in a grassy area</i>
antonym	<i>A woman is putting make-up on</i>	<i>The woman is removing make-up</i>

Table 2: Examples of phrase alignments constructed by phrasal axiom injection.

	Prec.	Rec.	Acc.
GRU	–	–	87.1
PL+eclassif	–	–	85.1
SemEval2014 Best Score	81.6	81.9	84.6
The Meaning Factory	93.6	60.6	81.6
LangPro	98.0	58.1	81.4
UTexas	–	–	80.4
W2W+P2P	84.2	77.3	84.3
W2W	97.1	63.6	83.1
P2P	85.6	72.1	83.0
No axioms	98.9	46.5	76.7

Table 3: RTE results on the SICK dataset.

paraphrases we extracted from the natural deduction proof in the training set. In particular, the examples of verb phrases show our method has the potential to capture long paraphrases. Each paraphrase in Table 2 is not contained in WordNet and PPDB. There are many instances of non-contiguous phrases in the SICK dataset, in particular, verb-particle phrases. Shown in Table 2, our semantic alignment can detect non-contiguous phrases through the variable unification process, which is one of the main advantages over other shallow/syntactic methods. In addition, Table 2 shows our method is not limited to hypernym or hyponym relations, but it is also capable for detecting antonym phrases.

5.4 RTE evaluation results

Table 3 shows the experimental results. The results show that the combination of word abduction and phrase abduction improved the accuracy. When the **W2W+P2P** result is substituted for the **W2W** result, there is a 1.1% increase in accuracy

(from 83.1% to 84.3%). The accuracy of **P2P** is almost equal to that of **W2W**. This is because the recall improves from 63.6% to 72.1% while the precision decreases from 97.1% to 85.6%. The increase in false positive cases caused this result; some details of false positive cases are described in the next subsection. **W2W+P2P** outperformed other purely logic-based systems. The machine learning-based approaches outperform **W2W+P2P**, but unlike these approaches, parameter estimation is not used in our method. This suggests that our method has the potential to increase the accuracy by using a classifier.

5.5 Positive examples and error analysis

Table 4 shows some positive and negative examples on RTE with the SICK dataset. For ID 9491, the sentence pair requires the paraphrase from *a field of brown grass* to *a grassy area*, not included in previous lexical knowledges. Our phrasal axiom injection can correctly generate this paraphrase from a natural deduction proof, and the system proves the entailment relation.

ID 2367 is also a positive example of phrasal axiom injection. The phrasal axiom between *set fire to cameras* and *burn cameras with a blow torch* was generated. This example shows that our semantic alignment succeeds in acquiring a general paraphrase by separating logical expressions such as *some* from content words and also by accounting for syntactic structures such as the passive-active alternation.

For ID 3628, the axiom shown in the table was extracted from the following sentence pair with

ID	Sentence Pair	Gold	Pred	Axiom
9491	A group of four brown dogs are playing in a field of brown grass Four dogs are playing in a grassy area	Yes	Yes	$\forall x_1(\mathbf{field}(x_1) \wedge \mathbf{brown}(x_1) \wedge \mathbf{grass}(x_1)) \rightarrow \mathbf{grassy}(x_1) \wedge \mathbf{area}(x_1))$
2367	A person is burning some cameras with a blow torch The person is setting fire to the cameras	Yes	Yes	$\forall x_1 \forall y_1(\mathbf{burn}(y_1) \wedge \mathbf{with}(y_1, x_1) \wedge \mathbf{blow_torch}(x_1) \wedge \mathbf{camera}(\mathbf{obj}(y_1))) \rightarrow \mathbf{set}(y_1) \wedge \mathbf{fire}(\mathbf{obj}(y_1)) \wedge \mathbf{to}(y_1, \mathbf{obj}(y_1)) \wedge \mathbf{camera}(\mathbf{obj}(y_1)))$
3628	A pan is being dropped over the meat The meat is being dropped into a pan	Unk	Yes	$\forall y_1(\mathbf{pan}(\mathbf{obj}(y_1)) \rightarrow \mathbf{into}(y_1, \mathbf{obj}(y_1)))$
96	A man is jumping into an empty pool There is no biker jumping in the air	Unk	No	$\forall y_1(\mathbf{jump}(y_1) \rightarrow \exists x_1(\mathbf{in}(y_1, x_1) \wedge \mathbf{air}(x_1)))$ $\forall y_1(\mathbf{man}(y_1) \rightarrow \mathbf{biker}(y_1))$
408	A group of explorers is walking through the grass Some people are walking	Yes	Unk	

Table 4: Positive and negative examples on RTE from the SICK dataset.

their *entailment* label:

T_1 : A woman is putting meat in a pan

H_1 : Someone is dropping the meat into a pan

But the phrase *drop over* does not entail the phrase *drop into*, and a proof for the inference is over-generated in ID 3628. We extracted all possible phrasal axioms from the training dataset, so noisy axioms can be extracted as a consequence of multiple factors such as parsing errors or potential disambiguation in the training dataset. One possible solution for decreasing such noisy axioms would be to use additive composition models (Tian et al., 2016) and asymmetric learnable scoring functions to calculate the confidence on these asymmetric entailing relations between phrases.

ID 96 is also an example of over-generation of axioms. The first axiom, $\forall y_1(\mathbf{jump}(y_1) \rightarrow \exists x_1(\mathbf{in}(y_1, x_1) \wedge \mathbf{air}(x_1)))$ was extracted from the proof of $T_1 \Rightarrow H_1$:

T_1 : A child in a red outfit is jumping on a trampoline

H_1 : A little boy in red clothes is jumping in the air

The second axiom $\forall y_1(\mathbf{man}(y_1) \rightarrow \mathbf{biker}(y_1))$ was extracted from the proof of $T_2 \Rightarrow H_2$:

T_2 : A man on a yellow sport bike is doing a wheelie and a friend on a black bike is catching up

H_2 : A biker on a yellow sport bike is doing a wheelie and a friend on a black bike is catching up

Although these axioms play a role in the proofs of $T_1 \Rightarrow H_1$ and $T_2 \Rightarrow H_2$, the wrong axiom $\forall y_1(\mathbf{man}(y_1) \rightarrow \mathbf{biker}(y_1))$ causes the over-generation of a proof for the inference in ID 96. The correct one would rather be $\forall x_1 \forall y_1(\mathbf{man}(y_1) \wedge \mathbf{on}(y_1, x_1) \wedge \mathbf{bike}(x_1) \rightarrow \mathbf{biker}(y_1))$. In this case, it is necessary to bundle predicates in a noun-phrase by specifying the types of a variable (entity or event) when making phrase alignments.

For ID 408, the word *explorer* is not contained in the training entailment dataset and hence the relevant axiom $\forall x_1(\mathbf{explorer}(x_1) \rightarrow \mathbf{people}(x_1))$ was not generated. While our logic-based method enables detecting semantic phrase correspondences in a sentence pair in an unsuper-

vised way, our next step is to predict unseen paraphrases of this type.

6 Conclusion

In this paper, we proposed a method of detecting phrase correspondences through natural deduction proofs of semantic relations between sentence pairs. The key idea is to attempt a proof with automatic phrasal axiom injection by the careful management of variable sharing during the proof construction process. Our method identifies semantic phrase alignments by monitoring the proof of a theorem and detecting unproved sub-goals and logical premises. The method of detecting semantic phrase alignments would be applicable to other semantic parsing formalisms and meaning representation languages such as abstract meaning representations (AMR) (Banarescu et al., 2013). Experiment results showed that our method detected various phrase alignments including non-contiguous phrases and antonym phrases. This result may contribute to previous phrase alignment approaches. The extracted phrasal axioms improved the accuracy of RTE tasks.

In future work, we shall enhance this methodology of phrasal axiom injection to predict unseen paraphrases. The pairs of premises and sub-goals that can be detected through the proof process conduct semantic alignments in a sentence pair. With the use of an additive composition model of distributional vectors, we can evaluate the validity of such semantic alignments. A combination of our phrasal axiom injection and additive composition model of distributional vectors has the potential to detect unseen paraphrases in a sentence pair.

Acknowledgments

We thank the three anonymous reviewers for their detailed comments. This work was supported by JST CREST Grant Number JPMJCR1301 and AIP Challenge Program, Japan.

References

- Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2492–2502.
- Lasha Abzianidze. 2016. Natural solution to FraCaS entailment problems. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Berlin, Germany, pages 64–74.
- Yuki Arase and Jun'ichi Tsujii. 2017. Monolingual phrase alignment on parse forests. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1–11.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, Sofia, Bulgaria, pages 178–186.
- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 597–604.
- Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets Markov: Deep semantics with probabilistic logical form. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*. Atlanta, GA, pages 11–21.
- Islam Beltagy, Stephen Roller, Gemma Boleda, Katrin Erk, and Raymond J. Mooney. 2014. UTexas: Natural language semantics using distributional semantics and probabilistic logic. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 796–801.
- Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. 2016. Representing meaning with a combination of logical and distributional models. *Computational Linguistics* 42(4):763–808.
- Yves Bertot and Pierre Castran. 2010. *Interactive Theorem Proving and Program Development: Coq'Art The Calculus of Inductive Constructions*. Springer Publishing Company, Incorporated, New York, USA.
- Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The Meaning Factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 642–646.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 632–642.
- Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Barcelona, Spain, pages 33–40.
- Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics* 33(4):493–552.
- Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspers, Hans Kamp, Manfred Pinkal, Massimo Poesio, Stephen Pulman, et al. 1994. FraCaS—a framework for computational semantics. *Deliverable D6*.
- Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. 2016. Fifty years of graph matching, network alignment and network comparison. *Information Sciences* 346-347(Supplement C):180 – 197.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology*. Association for Computational Linguistics, Atlanta, Georgia, pages 758–764.
- Bevan Keeley Jones. 2016. *Learning words and syntactic cues in highly ambiguous contexts*. Ph.D. thesis, The University of Edinburgh.
- Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 329–334.

- Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Doha, Qatar, pages 990–1000.
- Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. The Association for Computational Linguistics, pages 1077–1086.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*. European Language Resources Association, Reykjavik, Iceland, pages 216–223.
- Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of the 2016 System Demonstrations of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany, pages 85–90.
- Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 710–720.
- George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38(11):39–41.
- Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2055–2061.
- Koji Mineshima, Ribeka Tanaka, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2016. Building compositional semantics and higher-order inference system for a wide-coverage japanese CCG parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2236–2242.
- Terence Parsons. 1990. *Events in The Semantics of English: a Study in Subatomic Semantics*. MIT Press, Cambridge, USA.
- Dag Prawitz. 1965. *Natural Deduction – A Proof-Theoretical Study*. Almqvist & Wiksell, Stockholm, Sweden.
- Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics* 2:377–392.
- Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatakowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics* 4:127–140.
- Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Learning semantically and additively compositional distributional representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany, pages 1277–1287.
- Anne Troelstra and Helmut Schwichtenberg. 2000. *Basic Proof Theory*. Cambridge University Press.
- Hitomi Yanaka, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki. 2017. Determining semantic textual similarity using natural deduction proofs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 692–702.
- Wenpeng Yin and Hinrich Schütze. 2017. Task-specific attentive pooling of phrase alignments contributes to sentence matching. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 699–709.
- Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Vancouver, Canada, pages 277–287.
- Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve SMT-based paraphrasing model. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Columbus, Ohio, pages 1021–1029.