

# 共参照解析のための事象間関係知識の文脈化\*

井之上 直也<sup>†,‡</sup>

杉浦 純<sup>‡</sup>

乾 健太郎<sup>‡</sup>

<sup>†</sup>(株)デンソー基礎研究所

<sup>‡</sup>東北大学 大学院情報科学研究科

naoya\_a\_inoue@denso.co.jp

{naoya-i, jun-s, inui}@ecei.tohoku.ac.jp

## 1 はじめに

計算機の記事理解では、常識的知識（事象間の因果関係など）のデータベースが重要な役割を果たす。近年、大規模な Web コーパスから事象間の因果関係やスクリプト知識といった常識的知識を自動獲得する研究が盛んに行われ [2, 14, etc.], 言語解析への応用が始まっている [7, 13, etc.]。

同時に、獲得された知識の有効性を経験的に確認するためのデータセットの整備も進んでおり、そのうちの一つに Winograd Schema Challenge (WSC) [8] がある。WSC は、twin sentences と呼ばれるよく似た 2 文の対から構成される代名詞照応解析の問題集である。このデータセットは、(1) 人間には簡単に解ける、(2) 従来の統語的素性・意味的素性（述語の選択選好性等）の手がかりでは解けない、難しい照応解析の事例の問題集であるという特徴がある\*<sup>1</sup>。例えば、次のような事例が存在する：

(1) *John<sub>i</sub> fired Dave<sub>j</sub> because he<sub>j</sub> had too many absences.* (he=Dave)

(2) *John<sub>i</sub> fired Dave<sub>j</sub> because he<sub>i</sub> had too many employees.* (he=John)

これら 2 文は、同一の文構造であり、違いは *absences* か、*employees* のみである。he には John も Dave も入りうるため、統語的な特徴および意味的な特徴は役に立たない。ここで重要な手がかりとなるのは、「fire する側は従業員を抱える立場」「欠勤が多い人は fire される」といった、事象間の関係知識である。

以上のように、事象間の関係知識を有効利用し、その有効性を確認できる土壌が整ってきたものの、獲得された知識により言語解析の性能が飛躍的に向上したという報告は未だにない。例えば Rahman ら [13] は、文献 [2] の手法によって獲得された Narrative Schema (スクリプト知識) を用いて WSC の問題を解いているが、性能の向上はほとんどみられない\*<sup>2</sup>。

獲得した知識が言語解析の性能向上に寄与しない原因は、知識の規模の問題と知識の使い方の問題の二種類の問題が考えられる。我々の予備調査 [16] によれば\*<sup>3</sup>、知識の大規模化を行っても（従来比約 500 倍）依然とし

て知識の使い方の問題が残っており、性能向上を妨げていることが確認されている。

そこで本稿では、知識の使い方の問題に着目し、解決策について論じる。より具体的には、知識獲得の従来研究において、周辺文脈が捨象されていることを問題点として考え、知識を周辺文脈付きで獲得し、知識適用時の文脈との一致度を考慮した知識獲得・知識適用の枠組みについて調査する。本研究の貢献は、これまでにない規模の事象間関係知識（約 2 億インスタンス）を実際に代名詞照応解析に適用し、知識獲得時の周辺文脈の考慮の重要性を経験的に示したことにある。

本稿では、まず知識獲得の従来研究を概観し、問題点を述べる（2 節）。次に、この問題点を解消するために、周辺文脈付きの知識獲得・知識適用の枠組みを述べる（3 節）。最後に、WSC を通じて文脈考慮の有効性を検証した結果を述べる（4 節）。

## 2 関連研究

従来の知識獲得研究の対象となっている事象間関係知識の形式は、主に (a) 述語間関係 (e.g., *purchase ~ acquire*) [3, 6], (b) 変項付き述語間関係 (e.g., *purchase X ~ acquire X*) [9, 2], (c) タイプ付けされた変項付き述語間関係 (e.g., *purchase X ~ acquire X, X = COMPANY, COMPANY は X に入りうる名詞のクラス*) [12, 1, 10], の 3 種類に概ね分類できる（以後、これらを推論規則と呼ぶ。尚、本稿では、述語間の関連を示す記号として“~”を用いる）。また、発展途上ではあるが、文献 [15] のように STRIPS 形式を知識表現の方法として採用し、事象の前提条件や効果等、知識適用の条件を獲得するような研究もある。

しかしながら、これらの知識表現では、知識獲得時の述語の周辺文脈が捨象されてしまうため、獲得された知識を未観測の文章に適用する際に不適切な推論を行ってしまうという問題がある（より詳細な分析は、文献 [16] を参照されたい）。特に、(a) 述語の意味が多義である場合や、(b) 述語の意味は十分特殊だが、共有項以外の情報を伴わないと因果関係が成立しないような場合、(c) 因果関係が述語のモダリティを含めて初めて成立するような場合に、大きな問題となる。

例えば、“*The computer could not run the program because it has many errors.*” という文から現状の技術により獲得される推論規則は、*runX ~ Xhave, X = PROGRAM* である。しかし、この推論規則は次の二点で問題がある：(1) *have* の目的語 *error* が捨象されている、(2) *run* のモダリティ *could not* が捨象されている。重要なのは、「エラーがある」と「プログラムが動かない」という関係を抽出することであり、従来研究の知

\*Naoya Inoue, Jun Sugiura, and Kentaro Inui. Contextualizing Causal Knowledge for Coreference Resolution.

<sup>1</sup><http://www.cs.nyu.edu/davise/papers/WS.html>

<sup>2</sup>論文では Narrative Schema による性能向上が見られるが、誤記であることを確認した (personal communication)。

<sup>3</sup>約 7 億文の Web コーパスから獲得したスクリプト知識を用いて WSC に対する解析を行い、エラー分析を行った。詳細な報告は、文献 [16] を参照されたい。

表 1: 推論事例データベース (約 2 億エントリ) の例

述語:共有項役割 1	述語:共有項役割 2	共有項インスタンス	述語 1 の周辺文脈	述語 2 の周辺文脈
<i>run:dojb</i>	<i>have:nsubj</i>	program	aux:can, neg:not, nsubj:computer	dojb:error
<i>run:dojb</i>	<i>have:nsubj</i>	program	aux:could, neg:not, nsubj:machine	dojb:problem
<i>run:dojb</i>	<i>have:nsubj</i>	car	nsubj:I prep_with:friend	dojb:engine
<i>run:dojb</i>	<i>have:nsubj</i>	SUV	aux:could neg:not	neg:not dojb:fuel
<i>run:dojb</i>	<i>have:nsubj</i>	car	nsubj:I prep_with:friend	dojb:engine

識表現では不十分であると考えられる。我々の研究は、従来の知識獲得の研究において、知識獲得時の周辺文脈が捨象されたことにより知識をうまく使いこなせていないという仮説を立て、改善を図るものである。文献 [15] は、事象の前提条件等がある種の周辺文脈と捉えることもできるが、本研究は、事象を表現する述語自身の周辺文脈をより精緻に見るといって方向性が異なる。

### 3 文脈付き推論規則の自動獲得と適用方法

本稿では、2 節で述べた従来研究の問題点を解消するために、知識獲得時の文脈により条件付けされた推論規則 (文脈付き推論規則; *contextualized inference rules*) の知識表現と利用の一手法を示し、知識獲得・知識適用における文脈考慮の重要性について論じる (この主張の経験的な評価については 4 節を参照されたい)。本稿では、推論規則の知識表現手法として、タイプ付けされた変項付き述語間関係を出発点とし、議論を進める。

まず、従来研究の問題点を解消するための大方針として、推論規則の獲得と適用のそれぞれのフェーズを、次のような方針で拡張することを提案する: (a) 知識獲得時: 述語間の関係と変項タイプに加えて、述語の周辺文脈を同時に獲得する。(b) 知識適用時: 適用対象の文章の文脈と、適用される推論規則の文脈情報を照合し、より高い文脈類似度を持つ推論規則が積極的に適用されるようにする。これらの拡張を実現するためには、少なくとも次のような問題に取り組む必要がある:

- 推論規則の文脈表現として何をを用いるか?
- 文脈を含めた推論規則を、どのように一般化するか?
- 適用される推論規則の文脈と、観測された文章の文脈の類似度を、いかに文脈非依存な知識の信頼度に反映するか?

それぞれの解決方法について、次の節で述べる。

#### 3.1 文脈付き推論規則の知識表現

Web 上の大規模な文書集合から自動獲得した知識には、さまざまな粒度の言語表現で示された知識が含まれる。このため、獲得できた知識を最大限に談話処理に活用するには、知識獲得、または知識適用の際に、言語表現の粒度の違いを上手く吸収する仕組みが必要であることは、古くから認知されてきた課題である。

広く議論されてきた解決策として、獲得した知識を、事前に一般化された形で保持するという方法がある。2 節で述べたように、一般化された知識表現の方法には様々な選択肢が考えられ、その方法論についても、クラスタリングや語彙概念構造などの言語理論にもとづいた手法など、多岐に渡る。しかし、最適な一般化の粒度を、知識獲得の段階で一意に決めようとするのは困難であり、実際に、知識獲得研究者の間でコンセンサスを得られた一般化の粒度は未だに存在していない。

そこで本稿では、知識獲得時の文脈考慮の重要性を確認することに重きを置くために、できるだけ透明性が高く、かつ文脈利用の効果の上限値を確認できるようなアプローチとして、知識獲得時には一切一般化を行わずに、知識適用時に類似した事例を探索するアプローチ<sup>\*4</sup>を前提とし、知識表現を設計した。このアプローチは、知識適用のたびに巨大な推論事例の集合 (一般化された推論規則でなく、“事例”の集合である) を取得する (*retrieve*) 必要があるため、実用的な速度は見込めないが、事前の一般化を一切行わないため、手法の挙動を観察しやすく、問題点がクリアになりやすいという利点がある。

以上のような方針に基づき、知識表現の方法として、獲得された事例を加工せず (述語の共有項部分の変項化も行わない) に、そのままデータベースに保持する手法を採用する。本稿では、獲得されたデータベースを推論事例データベースと呼び、各エントリを推論事例と呼ぶ。文脈の表現方法としては、依存構造木<sup>\*5</sup>における述語の 1-近傍要素を用いる。述語対 *run, have* に関する推論事例データベースのエントリの例を、表 1 に示す。以後、推論事例 *e* の持つ各フィールドを順に、 $e.P_1, e.R_1, e.P_2, e.R_2, e.S, e.C_{e.P_1}, e.C_{e.P_2}$  と標記する。また、周辺文脈  $e.C_i$  の依存関係  $r$  の要素を、 $e.C_i.r$  と標記する。

本稿では、文脈付き推論事例を獲得する手法として、文献 [2] の手法をベースとする。詳細は文献 [2] に委ねるが、アルゴリズムの概要は次のとおりである:

ステップ 1: 大規模文書集合の各文書に対して共参照解析を適用し、同一文書内で共参照チェーンを成す名詞の集合を同定する。

ステップ 2: 各名詞の集合について、名詞と依存関係にある述語 (支配述語; *governing predicate*) ・依存関係の対の集合 (e.g., {*kill:nsubjpass, die:nsubj, grieve:prep\_for*}) を抽出する。

ステップ 3: 集合の各要素をペアワイズに組み合わせ、それぞれの述語の共有項部分の名詞 (e.g., *soldier, people*) と近傍の依存関係 (e.g., *prep\_by:war*) を文脈として取得し、推論事例データベースに追加する。

なお、この知識表現を用いた知識獲得手法の今後の発展の方向性としては、推論事例すべてを保持しておくのではなく、共有項インスタンスと文脈表現を基準として推論事例を事前にクラスタリングし、推論規則として保持しておくことが考えられる。

#### 3.2 文脈付き推論規則の利用

3.1 節の手法により得られた推論事例データベースを適切に使いこなすには、どのような仕組みが必要だろうか。ここで、「知識を使いこなす」とは、文章において観

\*4 ある種の、オンライン一般化と考えることもできる。

\*5 本稿では、Stanford Dependency を採用した。

測された情報  $F$  と、推論事例  $F' \sim X$  を組合せて ( $F'$  は、 $F$  と類似した命題)、文章と矛盾せず、文脈に則した尤もらしい新情報  $X$  を導くこととする。

例えば、“*I couldn't run the program.*” という情報を観測した場合、表 1 の推論事例を組合せて、“*The program has errors.*” という情報を導くことを指す (以後、観測文章における知識の適用対象の述語を観測述語、推論事例の共有項にあたる部分を、観測スロットインスタンス、文脈を観測文脈と呼ぶ)。一方で、*I could run the software because it was easy to use.* のような文章から獲得された推論事例を適用し、“*The program is easy.*” という情報を導くのは適切な推論ではない。2 節で述べたように、従来研究の知識表現では文脈情報が捨象されているため、このような推論が起きてしまうことが考えられる。

我々の分析によると [16]、推論規則の適用誤りの原因として、(a) 共有項部分の不一致 (e.g., 表 1 の car, SUV の推論事例を前述の例に適用してしまう)、(b) 文脈の不一致、の少なくとも二種類が存在し、エラーのマジョリティを占めることが分かっている。

推論事例データベースを利用してこのような不適切な推論を除外するために、知識適用時のスコアを、文脈非依存のスコア  $Score_{indep}$  と、文脈依存のスコア  $Score_{app}$  の二種類のスコアの積  $Score_{indep} \cdot Score_{app}$  により表現する。文脈非依存のスコアについては、本研究では、文献 [1] と同様に、〈述語, 役割〉間の相関を自己相互情報量により計算する (e.g.,  $\log \frac{P(\text{run:do}bj, \text{have:nsubj})}{P(\text{run:do}bj)P(\text{have:nsubj})}$ ); 詳細は、文献 [1] を参照されたい。

文脈依存のスコアについては、(a) 推論事例の共有項インスタンスと観測スロットインスタンスの類似度の計算と、(b) 推論事例の文脈と観測文脈の類似度を計算する。いま、観測述語  $P_t$  と依存関係  $R_t$  にある観測スロットインスタンス  $S_t$ 、依存構造木上での  $P_t$  からの 1-近傍要素の集合を  $C_t$  (観測文脈) とする。このとき、推論事例  $e$  を  $P_t$  に適用する際の文脈依存スコア  $Score_{app}(e, P_t)$  を、次のように定義する:

$$Score_{app}(e, P_t) = sim_S(e.S, S_t) \cdot sim_C(e.C_{P_t}, C_t) \quad (1)$$

以下、それぞれの類似度関数を説明する。

項の類似度 類似度  $sim_S$  については、各項から得られる単語ベクトル間のコサイン類似度を用いる。すなわち、 $sim_S(S_i, S_j) = \text{cosim}(v_{S_i}, v_{S_j})$  とする。

文脈の類似度 最後に、文脈間の類似度  $sim_C$  については、各依存関係ごとに類似度を計算する。類似度の計算には、共有項の類似度計算と同様、依存関係にある単語から得られる単語ベクトル間のコサイン類似度を用いる:

$$sim_C(C_i, C_j) = \sum_{r \in \text{codep}(C_i, C_j)} w_r \cdot \text{cosim}(v_{C_{i,r}}, v_{C_{j,r}}) \quad (2)$$

ここで、 $\text{codep}(C_i, C_j)$  は、文脈  $C_i, C_j$  の両方に共通に出現する依存関係の集合、 $w_r$  は、文脈類似度における依存関係  $r$  の重要度である。本稿では、あらゆる  $r$  に対して  $w_r = \frac{1}{|\text{codep}(C_i, C_j)|}$  としたが、教師あり学習等により適切な重みを設定することも考えられる。

表 2: 文脈付き推論規則を用いた照応解析の性能

Model	$Score_{indep}$	$Score_{app}$	Accuracy
BASELINE	-	-	56.3 (178/316)
INDEP	•		59.2 (187/316)
TYPED	•	$sim_S$	60.4 (191/316)
CONTEXT	•	$sim_C$	65.2 (206/316)
T&C	•	$sim_S, sim_C$	<b>65.5 (207/316)</b>

## 4 評価

本節では、知識適用時の文脈考慮の効果を調べるために、推論事例データベースの推論事例を Winograd Schema Challenge [8] に応用した結果を報告する\*6。推論事例データベースの構築には、ClueWeb 2012 コーパス\*7の約 1/4 (約 2.2 億文書、6.9 億文) を Stanford CoreNLP\*8 により解析し、同一文内、および隣接文間に存在する共参照関係を抽出した。この結果、約 2.3 億個の推論事例\*9からなる推論事例データベースが構築された。単語間の類似度計算には、Google により公開されている単語ベクトルデータベース\*10を用いた。このデータベースは、300 万語の単語・句について、その意味を分布により表現した 300 次元のベクトルを格納している。各ベクトルは、大規模な新聞記事コーパス (1,000 億単語) の上で、Skip-gram モデルアーキテクチャ[11] により学習された結果である。

### 4.1 Winograd Schema Challenge への応用

構築した推論事例データベースを代名詞照応解析に応用するために、代名詞  $p$  とその先行詞候補の集合  $A_p = \{a_i\}_{i=1}^{N_p}$  のペア  $\langle p, A_p \rangle$  を入力として、最尤の先行詞候補  $a_{p^*}$  を返す  $N_p$  値分類器を構築する。分類モデルには、 $k$ -近傍法を用い、入力  $\langle p, A_p \rangle$  から近傍  $k$  個の推論事例の集合  $E(\langle p, A_p \rangle, k)$  により、最も多く項共有を示唆されている  $a_i$  を最尤先行詞として選択する。アイデアとしては、文献 [17, 13] で示されているような、述語間の項共有スコアの利用に概ね該当する。

$k$  近傍法で用いる入力  $\langle p, A_p \rangle$  と推論事例  $e$  の距離関数  $D(\langle p, A_p \rangle, e)$  は、3.2 節で定義したスコア関数を用いて次のように定義する:

$$D(\langle p, A_p \rangle, e) = \sum_{i=1}^{N_p} \Gamma(p, a_i, e) \cdot Score_{indep}(e) \cdot sim_S(e.S, a_i) \cdot sim_C(e.C_{P_p}, C_p) \cdot sim_C(e.C_{P_{a_i}}, C_{a_i}) \quad (3)$$

ここで、 $\Gamma(p, a_i, e)$  は、 $p$  の支配述語とその依存関係、 $a_i$  の支配述語とその依存関係のペアに対して推論事例  $e$  を適用可能ならば 1、それ以外は 0 を返す関数である。また、 $P_p, P_{a_i}$  は、それぞれ代名詞  $p$  の支配述語、先行詞候補  $a_i$  の支配述語を指し、 $C_p, C_{a_i}$  はそれぞれ  $p$  の周辺文脈と  $a_i$  の周辺文脈を指す。

直感的には、代名詞の支配述語と、全ての先行詞候補の支配述語を組合せて得られる近傍  $k$  個の推論事例によ

\*6 <http://www.hlt.utdallas.edu/vince/data/emnlp12/>

\*7 <http://lemurproject.org/clueweb12/>

\*8 <http://nlp.stanford.edu/software/corenlp.shtml>

\*9 述語・項共有役割のペアで unique をとったところ、その個数は約 1,500 万であった。

\*10 <https://code.google.com/p/word2vec/>

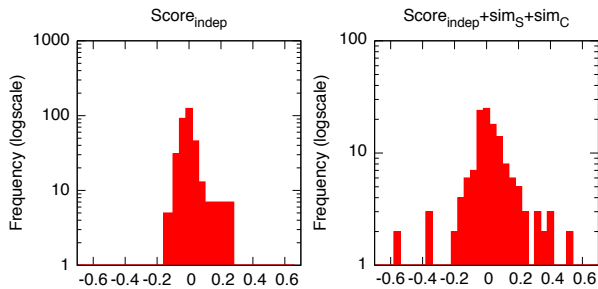


図 1: 各モデルの正例と負例に対するスコアの差の分布

る、先行詞候補への重み付き投票を行い、最尤先行詞を決定していることになる。

また、ベースラインモデルとして、各候補について支配述語と代名詞の支配述語のペアから取得した推論事例の集合の数をカウントし、最も数が多い候補を最尤先行詞として選択するモデル BASELINE を構築した。

#### 4.2 実験結果

照応解析の正解率による評価結果を、表 2 に示す。なお、本評価は、全 546 問中、文脈を考慮することにより改善が見込める事例、すなわち、正解の先行詞候補と不正解の先行詞候補の両方に対して、推論事例が見つかった問題 316 問を対象とした。表 2 より、文脈依存のスコアは正解率向上に大きく寄与した一方で、共有項の類似度スコアは、ほとんど正解率の向上に寄与しないことがわかった。

また、本稿で述べた文脈依存のスコアが、正しい先行詞（正例）と誤った先行詞（負例）の識別能力をどれだけ向上させるかを調べるために、（正例のスコア-負例のスコア）の分布を、図 1 に示した。図 1 より、文脈依存スコアを考慮することで正の差の分布（0.2 ~ 0.5 付近）に広がり生まれ、大きく識別能力を高められていることが確認できた。

以上の結果より、自動獲得した知識を談話処理で活用するためには、知識の規模の大きさに加えて、それをうまく使いこなす工夫が重要だということがわかった。また、INDEP、TYPED のそれぞれは、知識獲得の先行研究である文献 [9, 2]、文献 [1, 10] の知識表現に該当するが、これらの知識表現では、大規模化を行うだけでは性能向上に十分寄与しないことを示唆する結果が得られた。

最後に、手法の tractability を調べるために、パラメタ  $k$  に対する正解率の変化を、図 2 にプロットした。いずれの手法も、比較的小さい  $k$  において正解率のピークが確認された（TYPED については  $k = 3$ 、CONTEXT と T&C については  $k = 10$  付近）。本稿で用いた推論事例データベースでは、入力に対して適用可能な推論事例の集合  $E(\langle p, A_p \rangle, k)$  の要素数は全問題を通して平均 15,840<sup>\*11</sup>であったが、これらの事例との類似度を全て計算せずとも、近傍数個の推論事例との類似度だけを計算できればある程度の性能が出せることになる。近傍の推論事例の探索には、Locality Sensitive Hashing[5] などの近傍探索手法を用いることで、高速化が可能である。

\*11 正しい先行詞候補から平均 21,841、誤った先行詞候補から平均 9,838 の推論事例が取得された。

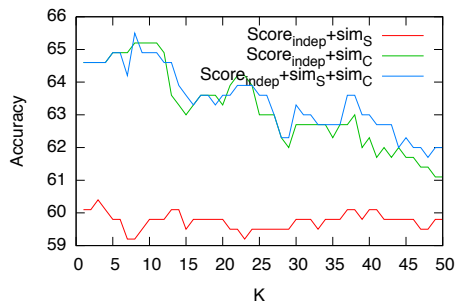


図 2: パラメタ  $k$  と正解率の関係

## 5 おわりに

これまでの談話処理において、自動獲得された知識を活用できていないことの原因として、知識の使い方の問題に着目し、その解決の指針を示した。より具体的には、先行研究の知識表現には、知識獲得時の文脈が捨象されているという問題点があることを指摘し、知識獲得時と知識適用時の両方で文脈を考慮することで、知識をより有効に使いこなせることを大規模知識の上で経験的に示した。今後は、獲得した推論事例データベースを一般化するために、共有項インスタンスと述語の周辺文脈の類似性を基準として、推論事例のクラスタリングを行う予定である。また、今回取り組んだ周辺文脈捨象の問題は、外部知識に依存した意味モデル（述語の選択選好性モデル等）にも共通であるため、推論事例の一般化の方向性と併せて、課題の整理を進めていく予定である。

謝辞 本研究は、JST 戦略的創造研究推進事業 CREST の一環として行われた。

## 参考文献

- [1] N. Chambers and D. Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL*, pp. 602–610, 2009.
- [2] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, pp. 789–797, 2008.
- [3] T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In D. Lin and D. Wu, editors, *Proceedings of EMNLP*, pp. 33–40. ACL, 2004.
- [4] I. Dagan, B. Dolan, B. Magnini, and D. Roth. Recognizing textual entailment: Rational, evaluation and approaches - Erratum. *Natural Language Engineering*, Vol. 16, No. 1, p. 105, 2010.
- [5] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, Vol. 99, pp. 518–529, 1999.
- [6] Andrew Gordon and Reid Swanson. Identifying personal stories in millions of weblog entries. In *Third International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA*, 2009.
- [7] J. Irwin, M. Komachi, and Y. Matsumoto. Narrative schema as world knowledge for coreference resolution. In *The Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 86–92, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- [8] H. J. Levesque. The Winograd Schema Challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.
- [9] D. Lin and P. Pantel. DIRT: discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference*, pp. 323–328, 2001.
- [10] Oren Melamud, Jonathan Berant, Ido Dagan, Jacob Goldberger, and Idan Szpektor. A two level model for context sensitive inference rules. In *ACL*, pp. 1331–1340, 2013.
- [11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.
- [12] Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H Hovy. Isp: Learning inferential selectional preferences. In *HLT-NAACL*, pp. 564–571, 2007.
- [13] A. Rahman and V. Ng. Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In *Proceedings of EMNLP-CoNLL*, pp. 777–789, 2012.
- [14] Tomohide Shibata and Sadao Kurohashi. Acquiring strongly-related events using predicate-argument co-occurring statistics and case frames. In *JCNLP*, pp. 1028–1036, 2011.
- [15] Avirup Sil and Alexander Yates. Extracting strips representations of actions and events. In *RANLP*, pp. 1–8, 2011.
- [16] 杉浦純, 井之上直也, 乾健太郎. 談話処理における語彙知識適用時の課題の整理. 言語処理学会第 20 回年次大会予稿集, p. To appear, 2014.
- [17] 飯田龍, 徳永健伸. 述語対の項共有情報を利用した文間セロ照応解析. 述語対の項共有情報を利用した文間セロ照応解析, pp. 804–807, 2010.