

# 複単語表現を利用した因果関係推定モデルの改善

佐々木 翔大<sup>1,a)</sup> 高瀬 翔<sup>2,b)</sup> 井之上 直也<sup>1,c)</sup> 岡崎 直観<sup>1,d)</sup> 乾 健太郎<sup>1,e)</sup>

**概要：** 計算機の文章理解において、2文間の因果関係の自動推定技術は重要な技術の一つである。これまでの研究では、主に語彙統語パターンや参照関係に基づいて因果関係にある事象のペアを大規模コーパスから自動獲得し、因果関係を確率統計的に推定する枠組みが提案されてきた。しかしながら、多くの研究は単語を事象の基本単位としており、“tired”-“give up”などの複単語表現で表される因果関係を適切に捉えることができないという問題がある。そこで本研究では、複単語表現のリストを作成し、既存の因果関係モデルを拡張することにより、複単語表現で表される因果関係を適切に捉える手法を提案する。また、因果関係推定モデルのベンチマークである Choice of Plausible Alternatives (COPA) [1] において複単語表現リストの導入の効果を検証し、現時点での最高性能を達成したことを報告する。

**キーワード：** 常識推論, 知識獲得, 因果関係, 事象間関係, 分散表現, 複単語表現

## 1. はじめに

計算機の文章理解において、事象間の因果関係の自動推定は重要課題のうちの一つであるが、未だに十分な推定精度が得られていない困難な問題である。高精度な因果関係の自動推定技術の実現により、質問応答や対話生成など、様々な自然言語処理の応用分野の発展が期待される。事象間の因果関係の自動推定に関するこれまでの研究では、語彙統語パターンや参照関係に基づいて、大規模なコーパスから因果関係にある事象のペアを自動獲得し、確率統計的に因果関係を推定するものが主流である [2], [3], [4]。例えば、Luo ら [3] は原因と結果の単語ペア (“bedtime”-“sleep” など) をコーパスから大量に獲得し、単語ペアの因果関係の強さを確率統計的に推定した。また、Grandroth-Wilding ら [5] は、コーパスから大規模に獲得した原因-結果を示す述語項構造のペア ((John, call, Mary)-(Mary, answer, -) など) を用いて、因果関係の強さを推定するニューラルネットワークを訓練する手法を提案した。Sharp ら [4] は、単語のペアによって表される因果関係の強さを柔軟に推定するために、分散表現を因果関係の推定に応用する手法を提

案した。

しかしながら、これらの手法には、複数単語で意味をなす複単語表現によって表される因果関係を捉えることができないという問題点がある。例えば以下の2文の因果関係を推定するとしよう。

- (1) The celebrity *got out of* the limousine.  
Cameras flashed in his direction.

文中の *get out of* は「～から降りる」という意味をなすが、これらの表現を構成するそれぞれの単語は全く別の意味である。しかし、従来手法は基本的に1単語を事象の単位として扱っており、文の正しい意味を捉えて因果推定ができるとは言い難い。

そこで本稿では、複単語表現を事象の単位として因果関係の強さを推定する手法を提案し、複単語表現知識を明示的に用いることが、因果推定タスクを解く性能に、どの程度寄与するかを実験で評価した。具体的には、オンライン辞書 Wiktionary から複単語表現のリストを作成し、因果関係推定のベンチマークである Choice of Plausible Alternatives (COPA) [1] に適用した。具体的には、最高性能を達成した Luo ら [3] の手法、分散表現に基づく因果関係のモデル化を行った Sharp ら [4] の手法を拡張し、複単語表現を含む因果関係を適切に推定できるようなモデルを構築した。また、COPA を用いて複単語表現リストの導入効果を検証し、現時点での最高性能を達成したことを報告する。さらに、実験結果を分析し、残された課題、今後の展望について述べる。

<sup>1</sup> 東北大学

Tohoku University

<sup>2</sup> 東北大学 (現在は NTT コミュニケーション科学基礎研究所所属)  
Tohoku University (current affiliation: NTT Communication Science Laboratories)

a) sasaki.shota@ecei.tohoku.ac.jp

b) takase.sho@lab.ntt.co.jp

c) naoya-i@ecei.tohoku.ac.jp

d) okazaki@ecei.tohoku.ac.jp

e) inui@ecei.tohoku.ac.jp

## 2. 関連研究

近年、事象間の因果関係を自動的に推定する研究が盛んに行われている [2], [3], [4]. また, Chambers ら [2] に代表される, 典型的に共起する事象の集合であるスクリプト的な知識 [6] の獲得・利用という文脈においても, 近年盛んに研究が行われている [2], [5], [7], [8]. その主なアプローチは, (1) 語彙統語パターンや参照関係に基づいて大規模コーパスから因果関係にある事象ペアを自動獲得し, (2) 確率統計的に因果関係を推定するものである. 例えば, Chambers ら [2] は, 主語や目的語を共有する動詞の集合 (“A criminal committed a crime” - “Police arrested the criminal” など) を因果関係にある事象の集合と仮定し, これらを動詞-項の形 (“X commit” - “arrest X” など) に汎化した上で, 相互情報量 (Point-wise Mutual Information) に基づいて関係の強度推定を行った.

Pichota ら [7] は, コーパスから獲得した事象ペアを述語項構造に汎化し, リカレントニューラルネットワークの一種である Long-Short Term Memory (LSTM) を用いてその関係をモデル化した. また, Modi ら [8], Grandroth-Wilding ら [5] は, スクリプト的な知識をフィードフォワード型のニューラルネットによってエンコードする手法を提案した. しかし, これらの手法は, 述語項として名詞句の主辞のみを考慮しているため, 複単語名詞表現 (high five など) を含む因果関係を適切に推定できない.

Sharp ら [4] は原因と結果を表す文から単語の分散表現を学習し, 畳み込みニューラルネットワークにより因果関係の推定を行う手法を提案した. また, Pichota ら [9] は, 前処理を施さない生の英語文を入出力とする LSTM encoder-decoder モデル [10] を用いて因果推定を行った. しかし, これらの手法は複単語表現の考慮が因果推定の性能にどの程度寄与するかを評価することができていない.

## 3. 先行手法

本節では, まずはじめに我々の提案手法の基本となる Luo ら [3], Sharp ら [4] の手法について説明する.

### 3.1 Causal Strength

Luo ら [3] は, 単語で表される事象間の因果関係の強さを, 必要要因性と十分要因性によりモデル化する *Causal Strength* という指標を提案した. 形式的には, 原因事象を表す単語  $i_c$  と, 結果事象を表す単語  $j_e$  に対して (本稿では, 原因を表す単語に添字  $c$ , 結果を表す単語に添字  $e$  を付け, bedtime $_c$ , sleep $_e$  のように表現する), Causal Strength  $CS(i_c, j_e)$  は次のように定義される.

$$CS(i_c, j_e) = CS_{nec}(i_c, j_e)^\lambda CS_{suf}(i_c, j_e)^{1-\lambda} \quad (1)$$

ここで,  $CS_{nec}(i_c, j_e)$  は,  $i_c$  の  $j_e$  に対する必要要因性 ( $i_c$

の生起が  $j_e$  の生起にどの程度必要不可欠か),  $CS_{suf}(i_c, j_e)$  は,  $i_c$  の  $j_e$  に対する十分要因性 ( $i_c$  の生起が  $j_e$  の生起をどの程度引き起こしうるか) であり,  $\lambda$  はハイパーパラメタである.

$CS_{nec}, CS_{suf}$  は以下のように算出される.

$$CS_{nec} = \frac{p(i_c|j_e)}{p^\alpha(i_c)} = \frac{p(i_c, i_e)}{p^\alpha(i_c)p(j_e)} \quad (2)$$

$$CS_{suf} = \frac{p(j_e|i_c)}{p^\alpha(j_e)} = \frac{p(i_c, i_e)}{p^\alpha(i_c)p(j_e)} \quad (3)$$

ここで  $\alpha^{*1}$  はハイパーパラメタである. それぞれの確率  $p(i_c, i_e), p(i_c), p(i_e)$  は, 各事象  $i_c, i_e$  のコーパスでの出現頻度に基づいて, 次のように計算される. まず, 事前に定義した, 因果関係を表すマーカー (B because A など) (以後 causal cue と呼ぶ) を含む文の集合を大規模なコーパスから獲得する. 表 1 に Luo らが用いた 53 個の causal cue を示した. 例えば, causal cue である “because” を含む, 下記のような文が獲得される.

(2) John gave a high five to Bob [because] John was happy.

次に, causal cue が表す原因部分の単語の集合  $A$  と, 結果部分の単語の集合  $B$  より, 原因単語と結果単語の対を抽出する (causal cue によって, 文のどこを  $A, B$  とするか異なる). ここでは, 文中の “because” が causal cue であるので,  $A = \{\text{John, was, happy}\}, B = \{\text{John, gave, a, high, five, to, Bob}\}$  となる. 次に, それぞれの単語の集合に対して各種フィルタリング処理を行った. 具体的には, lemmatization と小文字化の適用, stop words の除去を行い, さらに, 内容語 (名詞, 動詞, 形容詞, 副詞) であり, かつ WordNet [11] に含まれている単語のみを取り出す処理を行った.\*2 最後に,  $A$  と  $B$  の直積を取り, 原因単語と結果単語の対の集合を得る (例えば,  $\{(\text{happy}_c, \text{John}_e), (\text{happy}_c, \text{give}_e), \dots\}$ ). 獲得された全ての文についてこの処理を行った後, 原因単語としての出現頻度 ( $p(i_c)$ ), 結果単語としての出現頻度 ( $p(j_e)$ ), およびペアとしての出現頻度を用いて ( $p(i_c, j_e)$ ), 上記の確率推定を行った.

次に, 2 文  $S_c, S_e$  が与えられた時,  $S_c$  が  $S_e$  を引き起こすことの妥当性のスコア  $Score(S_c, S_e)$  を計算する方法を示す. 基本方針としては,  $S_c$  と  $S_e$  の内容語のペアについて, 式 (1) で与えられる因果関係スコアを計算し, その和を取ることで文間の因果関係スコアを得る. まず,  $S_c$  と  $S_e$  の単語を全て lemmatize し, stop words を除外し, 内容語 (名詞, 動詞, 形容詞, 副詞) であり, かつ WordNet [11] に含まれる単語を抽出する.\*3. 以後, この処理で各文

\*1 Luo らは  $\alpha = 0.66$  とした. 本研究も同じ値を用いる.

\*2 なお, ノイズの軽減のために,  $A, B$  に含まれる最大単語数は 10 単語に限定した.

\*3 元論文中には述べられていないが, さらにコーパス中での頻度 10 位以内の単語を除外する (personal communication).

表 1 53 Causal cues. A, B はそれぞれ原因, 結果の区間, DET は a/an/the/one を, BE は is/are/was/were を表す.

intra-sentence			inter-sentence		
A lead to B	A leads to B	A led to B	If A, then B	If A, B	B, because A
A leading to B	A give rise to B	A gave rise to B	B because A	B because of A	Because A, B
A given rise to B	A giving rise to B	A induce B	A, thus B	A, therefore B	B, A as a consequence
A inducing B	A induces B	A induced B	Inasmuch as A, B	B, inasmuch as A	In consequence of A, B
A cause B	A causing B	A causes B	B due to A	Due to A, B	B in consequence of A
A caused B	B caused by A	A bring on B	B owing to A	B as a result of A	As a consequence of A, B
A brought B	A bringing on B	A brings on B	A and hence B	Owing to A, B	B as a consequence of A
B result from A	B resulting from A	B results from A	A, hence B	A, consequently B	A and consequently B
B resulted from A	the reason(s) for/of B	BE A	A, for this reason alone , B		
DET effect of A	BE B	A BE DET reason(s) of/for B			

から得られた単語の集合を  $W(S_c), W(S_e)$  と呼ぶ.

最後に,  $W(S_c)$  と  $W(S_e)$  を利用し, 次のように文間の因果関係スコアを計算する.

$$Score(S_c, S_e) = \frac{1}{|W(S_c)| + |W(S_e)|} \sum_{i \in W(S_c)} \sum_{j \in W(S_e)} CS(i, j) \quad (4)$$

$|W(S_c)|, |W(S_e)|$  は文  $W(S_c), W(S_e)$  に含まれる単語の数である. なお, 本稿の実験では予備実験の結果より  $|W(S_c)| \times |W(S_e)|$  をスコア正規化に用いた.

### 3.2 Causal Embedding

Sharp ら [4] は, 単語のペアによって表される因果関係の強さを柔軟に推定するために, 分散表現を因果関係の推定に応用する手法を提案した. 本手法では, まず 3.1 節の手法と同様に, 因果関係を表すマーカを利用して因果関係の記述を大規模コーパスから獲得し, これらを原因を表す単語 (原因単語) と結果を表す単語 (結果単語) のペアの集合  $D$  に分解する. (例えば,  $D = \{(\text{bedtime}, \text{sleep}), (\text{knock}, \text{open}), (\text{tired}, \text{rest}), \dots\}$ ) その後, Mikolov ら [12] により提案された Skip-gram モデルを利用し, ターゲット単語を原因単語, 文脈単語をその原因に対応する結果単語とし, 分散表現を学習する. より形式的には, 下記の目的関数を最大化する.

$$\sum_{(i_c, j_e) \in D} \log p(j_e | i_c) \quad (5)$$

ここで,  $p(j_e | i_c)$  は次のように定義される.

$$p(j_e | i_c) = \frac{1}{Z} \exp(v'_{j_e} \top v_{i_c}) \quad (6)$$

ここで,  $\frac{1}{Z}$  は正規化項,  $v'$  は Skip-gram における文脈単語ベクトル,  $v$  は Skip-gram におけるターゲット単語ベクトルである. 直感的には, 原因単語から, それに対応する結果単語を予測できるように, 各単語の分散表現  $v', v$  を学習することになる. 元の Skip-gram の学習では, 文脈単語ベクトル  $v'$  は捨てられてしまうが, 本手法では  $v'_w$  を結果

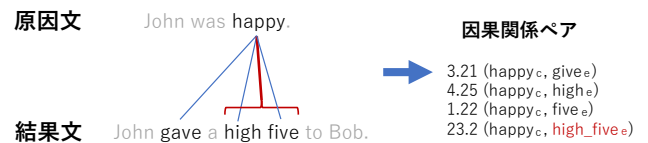


図 1 2 文間の因果関係推定手法の概要

単語  $w$  の分散表現,  $v_w$  を原因単語  $w$  の分散表現として推論時に利用する. より具体的には, 原因単語  $w_c$  と結果単語  $w_e$  の因果関係の強さを  $\text{cosim}(v_{w_c}, v'_{w_e})$  として推定する (ここで,  $\text{cosim}$  はコサイン類似度).

2 文間の因果関係の推定方法については, 元の論文では述べられていないが, 我々の実験では Luo ら [3] の計算手法, すなわち式 (4) と同様の計算式を用いた (ただし,  $CS(i, j)$  を  $v_i \top v'_j$  として計算した).

### 4. 提案: 複単語表現を利用した因果関係推定

3.1 節, 3.2 節で述べたような, 単語を事象の単位とする手法では, 複数単語で意味をなす複単語表現により表される因果関係を捉えることできないという問題点がある. 例えば例文 (2) では, “happy”-“high five” という関係を捉える必要があるが, 前述の手法では “happy”-“high”, “happy”-“five” のように, 単語毎の個別の関係に分解されてしまい, 正しく因果関係を捉えることができないと考えられる.

そこで我々は, 複単語表現のリストを作成し, 因果関係ペアの抽出処理において複単語表現を一つの単語とみなすことで, 3.1 節, 3.2 節の手法を複単語表現の意味を捉えられるように拡張する. 具体的には, 式 (4) における  $W(S_c), W(S_e)$  の集合に, 複単語表現が含まれるように因果関係ペアを抽出する. 例文 (2) に対する, 複単語表現を考慮した 2 文間の因果関係推定の手順を図 1 に示す. ここでは, これまでに抽出していた因果関係ペア ( $\text{happy}_c, \text{high}_e$ ), ( $\text{happy}_c, \text{five}_e$ ) などに加えて, 複単語表現を考慮した因果関係ペア ( $\text{happy}_c, \text{high\_five}_e$ ) を抽出している.

複単語表現のリストを作成するために, Wikipedia 形式の

ユーザ参加型の辞書作成プロジェクトである Wiktionary<sup>\*4</sup>を用いる。Wiktionary が持つエントリのうちでタイトルが  $n$  単語<sup>\*5</sup> からなり、見出しに Verb, Noun と記載があるエントリを複単語述語 (Multi Word Predicate; MWP), 複単語名詞 (Multi Word Noun; MWN) とみなし、リストに追加した。作成されたリストは、MWP, MWN がそれぞれ約 3 万個、約 10 万個の要素からなる。

## 5. 実験

複単語表現を考慮するモデルを利用することで、因果推定問題の性能がどれくらい向上するか、実験によって調査した。具体的には Luo らが提案した Causal Strength の手法と Sharp らが提案した Causal Embedding の手法で、複単語表現を考慮する場合と考慮しない場合の性能差を調べた。評価には因果関係推定のベンチマークである COPA[1]を用いた。

### 5.1 データ

因果関係ペアの抽出には ClueWeb12<sup>\*6</sup> を用いた。ClueWeb12 は Web ページをクロールすることで取得された約 7 億文書からなるコーパスである。この中から、causal cue を含む文を約 2 億文抽出し、取得した因果関係ペアの個数は単語のみを用いる手法で約 25 億、複単語表現を 1 単語とみなす提案手法では約 30 億個である。品詞タグ付け、lemmatization には Stanford Core NLP [13] を用いた。

また、モデルの評価には、Choice of Plausible Alternatives (COPA) [1]<sup>\*7</sup> を用いた。COPA は因果関係推定モデルのベンチマークセットであり、開発セット 500 問、テストセット 500 問の合計 1,000 問からなる。COPA の各問題は、前提文 1 文と選択肢となる候補文 2 文から成る問題で、前提文の原因あるいは結果としてより妥当な候補文を選ぶというタスクである。以下に COPA の例を示す。

前提文: The shirt shrunk.

What was the cause of this ?

候補文 1: I put it in the dryer.

候補文 2: I poured bleach on it.

この文では「シャツが縮んだ」ことの原因を問われており、正解は候補文 1「シャツを乾燥機に入れた」からである。なお、COPA のデータ 1,000 問のうち、本稿で作成した複単語表現リストに合致する MWP を含む問題は 41.2 % (412/1,000), MWN を含む問題は 32.4 % (324/1,000) であった。個数としては、MWP が 514 個、MWN が 461 個見つかった。

表 2 COPA の実験結果

手法	コーパス	正答率 (%)
CS-add $\lambda=1.0$ [3]	Causal Net	70.2
CS-mul $\lambda=0.8$ (ベースライン)	ClueWeb12	69.9
CS-mul w/mwp $\lambda=0.7$	ClueWeb12	<b>71.2</b>
CS-mul w/mwn $\lambda=0.5,0.7$	ClueWeb12	67.4, 68.4
CS-mul w/mwp, mwn $\lambda=0.7$	ClueWeb12	70.4
CE-mul (ベースライン)	ClueWeb12	68.6
CE-mul w/mwp	ClueWeb12	68.7
CE-mul w/mwn	ClueWeb12	63.9
CE-mul w/mwp, mwn	ClueWeb12	68.1

### 5.2 実験設定

まず、COPA を解く際の手順について説明する。前提文  $P$ , 候補文 1  $A_1$ , 候補文 2  $A_2$  が与えられたとき、妥当な結果を選択する問題については、 $Score(P, A_1)$  と  $Score(P, A_2)$  を用いて因果関係の強さを推定し (詳しくは式 (4) を参照), スコアが大きい方の候補文を回答として出力した。同様に、原因を選択する問題については  $Score(A_1, P)$  と  $Score(A_2, P)$  を計算し、妥当な候補文を選択した。stop words は自然言語処理ツールキットの NLTK<sup>\*8</sup> で定義されている単語のリストを用いた。

Causal Strength の手法におけるハイパーパラメータ  $\lambda$  は開発セットでチューニングを行った。実際に用いた値は表 2 に示す。Causal Embedding の手法においては、ベクトルの次元は 1000 次元とし、負例サンプリングに基づく学習 [12] を行った。負例サンプル時のサンプル数は 20 に設定し、エポック数は 5 とした。また、3.2 節で説明したとおり、Sharp ら [4] の分散表現学習では、Skip-gram におけるターゲット単語 ( $v$ ) を原因単語、文脈単語 ( $v'$ ) を結果単語として分散表現の学習を行っているが、その逆の設定、すなわちターゲット単語を結果単語、文脈単語を原因単語とする方法もあり得る。これらの二種類の設定で学習される分散表現は性質が違うものと考えられ、どちらの分散表現が COPA の問題に有効に働くかは自明でない。そこで本実験では、後者の設定で分散表現を学習する変形手法についても検討した。開発セットにおいて、原因を選択する問題に対しては Sharp らの手法が、結果を選択する問題に対しては前述の変形手法が最も良い性能を示したため、以降はこの設定を用いる。

### 5.3 実験結果

COPA のテストセット 500 問における正答率を表 2 に示した。ただし、表中の「-add」, 「-mul」はスコア正規化の項がそれぞれ  $|W(S_c)| + |W(S_e)|$ ,  $|W(S_c)| \times |W(S_e)|$  であることを表す。また、CS, CE はそれぞれ Causal Strength, Causal Embedding の手法を用いたことを表す。いずれの手法においても、MWP を考慮したモデルが MWP を考慮

<sup>\*4</sup> [https://en.wiktionary.org/wiki/Wiktionary:Main\\_Page](https://en.wiktionary.org/wiki/Wiktionary:Main_Page)

<sup>\*5</sup> 実験では  $n = 2, 3$  を用いた。

<sup>\*6</sup> <http://lemurproject.org/clueweb12/index.php>

<sup>\*7</sup> <http://people.ict.usc.edu/~gordon/copa.html>

<sup>\*8</sup> <http://www.nltk.org/>

しないモデルよりも良い性能に達し、MWP を考慮することが有効であることがわかった。表 2 中の 1 段目の正答率が Luo ら [3] によって報告されている従来手法における最高性能であるが、Causal Strength の手法で MWP を考慮したモデルは、これまでの COPA の最高性能を 1% 上回った。一方で、MWN を考慮したモデルは MWN を考慮しないモデルより性能が低下した。これは、MWN のコーパス中での因果関係ペアとしての登場回数が少なく（例えば (surprising party<sub>c</sub>, hide<sub>e</sub>) の登場回数は 2 回）十分な統計が取れないことが理由として考えられる。

複単語表現を用いた時の性能の改善の上限値を見積もるために、COPA の設問中の複単語表現を手手で同定し、そこで同定された複単語表現の辞書を用いて、同様に COPA をといた。ここでは Causal Strength の手法を用いた。その結果、最大 71.0% \*9 の正答率であった。この性能が、我々の最も性能の良いモデルと同等の性能であることから、複単語表現を用いることでのこれ以上の性能向上は難しいと考えられる。

#### 5.4 分析

はじめに、複単語表現を事象の単位とすることで、これまで解くことができていなかった問題が解けるようになった例を紹介する。

前提文: The father *shut off* the children's television.

What was the cause of this ?

候補文 1: It was bedtime for the children.

候補文 2: The children were watching cartoons.

前提文中の *shut* は 1 語では「(ドアなど)を閉める」という意味だが *shut off* という複単語表現で捉えると「(電気・機械など)を止める」という意味をなす。複単語表現を用いない手法では、前提文の中の複単語表現 *shut off* の意味を捉えることができず、不正解となっている。一方、MWP を用いる手法では *shut off* をひとまとまりで捉えて、正解することができている。実際に、Causal Strength の値をみると (bedtime<sub>c</sub>, shut<sub>e</sub>) の Causal Strength の値は 1.88 であるが、(bedtime<sub>c</sub>, shut off<sub>e</sub>) の値は 13.7 であり、他の因果関係ペアに比べて非常に高い値を示している。これは「bedtime (就寝時間)には電気などを *shut off* する(消す)」という因果関係を捉えていると言える。

また、不正解となった問題の中でも、複単語表現を考慮することにより、正解の候補文に含まれる事象に対する Causal Strength の値が実際に改善する事例があった。例を表 3 に示す。例えば *wait* が原因単語である時を考える。*take, seat* が単独で結果単語である時は、Causal Strength

表 3 複単語表現を考慮したことによる因果関係スコアの変化の例

原因の単語	結果の単語	CS
wait	take	3.28
wait	seat	2.38
wait	take a seat	12.9
sun	cast	3.95
sun	shadow	27.1
sun	cast a shadow	74.8
think	come	4.02
think	come up with	5.23

の値が 3.28, 2.38 と低い値となっているが、腰掛けるという意味の *take a seat* を結果の事象とした時は、Causal Strength の値は 12.9 と高い値になっている。つまり、複単語表現 *take a seat* を構成単語 *take, seat* それぞれ別に捉えてしまうと、*wait* との因果関係を捉えることができないが、*take a seat* をひとまとまりで事象の単位とすることで、*wait* との因果関係をより正確に捉えることができています。

## 6. おわりに

本稿では、Wiktionary から複数単語で意味を成す複単語表現のリストを作成し、既存の因果関係モデルを拡張することにより、複単語表現で表される因果関係を適切に捉える手法を提案した。また、因果関係推定モデルのベンチマークである COPA の評価において複単語表現辞書の導入の効果を検証し、現時点での最高性能を達成したことを報告した。今後の課題としては、因果関係の事象の単位を述語と目的語の組合せとすることなどが考えられる。

謝辞 本研究は JST CREST(課題番号: JPMJCR1513) の支援を受けて行った。本研究は JSPS 科研費 16H06614, 15H05318 の助成を受けたものである。本研究の実施にあたり、松林優一郎氏 (東北大学) にご助言を頂いた。ここに感謝の意を表す。

## 参考文献

- [1] Roemmele, M., Bejan, C. A. and Gordon, A. S.: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning., *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning* (2011).
- [2] Chambers, N. and Jurafsky, D.: Unsupervised Learning of Narrative Event Chains., *Proc. of ACL*, pp. 789–797 (2008).
- [3] Luo, Z., Sha, Y., Zhu, K. Q., won Hwang, S. and Wang, Z.: Commonsense Causal Reasoning between Short Texts, *Proc. of KR*, pp. 421–430 (2016).
- [4] Sharp, R., Surdeanu, M., Jansen, P., Clark, P. and Hammond, M.: Creating Causal Embeddings for Question Answering with Minimal Supervision, *Proc. of EMNLP*, pp. 138–148 (2016).
- [5] Granroth-Wilding, M. and Clark, S.: What Happens Next? Event Prediction Using a Compositional Neural Network Model, *Proc. of AAAI*, pp. 2727–2733 (2016).

\*9 今回は評価セットを対象に複単語表現の同定を行い、開発セットの複単語表現の同定を行っていない。従って開発セットにおけるハイパーパラメータのチューニングができなかったため、性能の最大値を示した。

- [6] Schank, R. C. and Abelson, R. P.: *Scripts, plans, and knowledge*, Yale University New Haven, CT (1975).
- [7] Pichotta, K. and Mooney, R. J.: Using Sentence-Level LSTM Language Models for Script Inference, *Proc. of ACL*, pp. 279–289 (2016).
- [8] Modi, A. and Titov, I.: Inducing Neural Models of Script Knowledge., *Proc. Of CoNLL*, Vol. 14, pp. 49–57 (2014).
- [9] Pichotta, K. and Mooney, R. J.: Learning Statistical Scripts with LSTM Recurrent Neural Networks., *Proc. of AAAI*, pp. 2800–2806 (2016).
- [10] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to sequence learning with neural networks, *Proc. of NIPS*, pp. 3104–3112 (2014).
- [11] Fellbaum, C.: *WordNet: An Electronic Lexical Database*. Bradford Books., MIT and Press (1998).
- [12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Proc. of NIPS*, pp. 3111–3119 (2013).
- [13] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. and McClosky, D.: The stanford corenlp natural language processing toolkit., *Proc. of ACL: System Demonstrations*, pp. 55–60 (2014).