

B3TB2099

卒業論文

複単語表現による因果関係推定モデルの改善

佐々木翔大

2017年 3月 31日

東北大学
工学部 情報知能システム総合学科

複単語表現による因果関係推定モデルの改善*

佐々木翔大

内容梗概

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要素技術の一つである。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、ほとんどの研究は単語を事象の基本単位としており、shut off-sleep 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、複単語表現の辞書を作成し、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する。また、実験結果を分析し、残された課題について述べる。

キーワード

複単語表現, 因果推定

*東北大学 工学部 情報知能システム総合学科 卒業論文, B3TB2099, 2017年3月31日.

Contents

1	はじめに	1
2	背景	2
2.1	関連研究	2
2.2	Causal Strength	2
2.3	Causal Embedding	4
3	提案手法	5
3.1	複単語辞書を利用した因果関係の獲得	5
3.2	複単語表現を考慮したことによる効果	6
4	実験	6
4.1	データ	7
4.2	実験設定	7
4.3	実験結果	8
4.4	分析	9
5	おわりに	11
	謝辞	12
	付録	15

List of Figures

List of Tables

1	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 . . .	6
2	COPA の実験結果	8
3	特定品詞を除外して COPA をといた時の正答率	9
4	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.	15

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する Causal Strength という指標を提案した。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。

しかしながら、これらの従来手法には、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。

The public figure got out of the limousine. Cameras flashed in his direction.

文中の「public figure」は「有名人」という意味、「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、先の手法は基本的に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。より具体的には、オンライン辞書 Wiktionary から複単語表現の辞書を作成し、因果関係推定のベンチマークである Choice of Plausible Alternatives (COPA) [1] で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辞書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。

2 背景

2.1 関連研究

冒頭で述べたように、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 [cite]. 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, 述語項を埋める名詞句を主辞に捨象してしまうため, 複単語名詞表現 (public figure など) を含む因果関係を適切に推定できないと考えられる. 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.

Luo らの手法 [3] と Sharp ら [4] の手法は実験でも用いるため 2.2, 2.3 章にて詳しく説明する

2.2 Causal Strength

Luo らは原因単語と結果単語のペアの因果関係の強さを表す Causal Strength の算出法を提案した.

はじめに, 彼らは原因単語 i_c と結果単語 j_e の組み合わせを causal pair(i_c, j_e)

と呼称し、事象の単位とした。また、因果関係は因果の必要性、十分性を加味しなければならないという直感を元に、causal pair (i_c, j_e) に対する Causal Strength を因果の必要性、十分性を表した CS_{nes} , CS_{suf} を用いて以下のように定義した。(ただし λ はハイパーパラメータである。)

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

CS_{nes} , CS_{suf} は以下のように算出される。

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

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

(ただし分母を α^1 乗した。) 例えば、(rainfall_c, flooding_e) という事象には、結果として flooding が起きる大抵の状況では、原因として rainfall が起きているという点で必要性が存在する。また、(storm_c, damage_e) という事象には、原因として storm が起きる大抵の状況では、結果として damage が起きているという点で十分性が存在する。この必要性、十分性をモデル化したものが、 CS_{nes} , CS_{suf} である。事象に必要性が強いほど、 CS_{nes} は大きくなり、十分性が強いほど、 CS_{suf} が大きくなる。

$p(i_c), p(j_e), p(i_c, j_e)$ は以下の式で計算する。

$$p(i_c) = \frac{\sum_{w_e \in W_e} f(i_c, w_e)}{M} \quad (4)$$

$$p(j_e) = \frac{\sum_{w_c \in W_c} f(w_c, j_e)}{M} \quad (5)$$

$$p(i_c, j_e) = \frac{f(i_c, j_e)}{N} \quad (6)$$

ただし、 M は

$$M = \sum_{i_c \in W_c} \sum_{j_e \in W_e} f(i_c, j_e) \quad (7)$$

と計算され、 N は causal pair の種類数である。 W_c, W_e は causal pair 内に出現する全ての単語 w_c, w_e の集合、 $f(i_c, j_e)$ は causal pair (i_c, j_e) の出現頻度である。

最後に $f(i_c, j_e)$ の計算方法、つまり causal pair (i_c, j_e) の数え上げ方について説明する。まず、事前に定義した、因果関係を表すマーカーである causal cue を含む文を大規模なコーパスから獲得する。以下の文を例に考える。

¹Luo らは $\alpha = 0.66$ とした。) 本研究も同様の値を用いる。

The storm [caused] tremendous flooding. (*)

文中の「caused」が causal cue であるので，原因単語と結果単語の区間 A, B が決定される．ノイズ軽減のために A, B に含まれる最大単語数は 10 単語に限定した．表 4 に Luo らが用いた 53 個の causal cue を示した．

区間 A に含まれる単語 i_c と区間 B に含まれる単語 j_e の組み合わせを causal pair (i_c, j_e) とみなし，共起頻度を計算する．ただし， i_c, j_e は lemmatize したものをを用いる．例文では $(storm_c, tremendous_e)$, $(storm_c, amount_e)$, $(storm_c, damage_e)$, $(storm_c, landing_e)$, $(storm_c, beach_e)$ を causal pair とみなし， $f(i_c, j_e)$ を計算する．この工程で stop words は除外し，さらに名詞，動詞，形容詞，副詞のいずれかであり，かつ WordNet に含まれる単語に限定した．

この計算過程からもわかるように，複数単語で意味をなす複単語表現の意味を捉えることができないという問題点がある．例えば，以下の文がコーパス中で出現したとする．

The man *got out of* the car [because] he arrived home. (**)

このとき「because」が causal cue なので，この文から抽出される causal pair は $(arrive_c, man_e)$, $(arrive_c, get_e)$, $(arrive_c, car_e)$, $(home_c, man_e)$, $(home_c, get_e)$, $(home_c, car_e)$ となり，文中の「get out of」の複単語表現は捉えることができていないことがわかる．

2.3 Causal Embedding

Sharp ら [4] は原因単語と結果単語のペアを用いて，原因単語と結果単語それぞれの分散表現 Causal Embedding を学習した．学習方法は Mikolov らが提案した Skip-gram モデル [11] をベースとしており，その中におけるターゲット単語を原因単語，文脈単語を結果単語と置き換えて考えることで，その他は Skip-gram と同様のアルゴリズムでベクトルを学習した．これによって (*) の文がコーパス中で登場した時の，学習の概要を図 2 に示した．

図のように原因単語「storm」のベクトルから結果単語「tremendous」「flooding」の単語ベクトルを予測するように学習する．具体的にはコーパス中で共起した原因単語と結果単語の内積が高まるように学習される．

学習における目的関数は以下である．

$$-\sum_{i_c \in W_c} \sum_{j_e \in W_e} \log P(j_e | i_c) \quad (8)$$

この条件付き確率を負例サンプル [11] で計算する.

$$\log P(j_e|i_c) = \log \sigma(v_{j_e}^\top v_{i_c}) + \sum_{i=1}^k \mathbf{E}_{j_{e_neg} \sim P(W_e)} [\log \sigma(-v_{j_{e_neg}}^\top v_{i_c})] \quad (9)$$

ただし, v_{i_c} , v_{j_e} は原因単語 i_c , 結果単語 j_e に対応するベクトル表現であり, k は負例サンプル数である.

この目的関数からわかるように, コーパス中でよく共起する単語ペアの内積が高まるように単語のベクトル表現が学習される. しかしながら, この手法においても, 複数単語で意味をなす複単語表現の意味を捉えることができないという, Causal Strength と同じ問題点がある.

3 提案手法

3.1 複単語辞書を利用した因果関係の獲得

複単語表現の意味を捉えることができない問題に対処するために我々は, 複単語表現の辞書を作成し, 複単語表現も事象の単位とみなす手法を提案する. 複単語表現の辞書を作成するために, Web 上の辞書である Wiktionary² を用いる. Wiktionary に記載されている記事でタイトルが n 単語³ からなり, 見出しに Verb, Noun と記載がある記事のタイトルを複単語述語 (MWP), 複単語名詞 (MWN) とみなし, 辞書に追加した. そして, コーパスからの causal pair 抽出時に, 辞書中にある複単語表現が登場した時, それも 1 単語としてみなし, 単語と同じように扱う.

(*) の文を考えると, get out of が複単語表現の辞書に存在するので, ここから得られる causal pair は $(arrive_c, man_e)$, $(arrive_c, get_e)$, $(arrive_c, car_e)$, $(arrive_c, get\ out\ of_e)$, $(home_c, man_e)$, $(home_c, get_e)$, $(home_c, car_e)$, $(home_c, get\ out\ of_e)$

²https://en.wiktionary.org/wiki/Wiktionary:Main_Page

³ $n = 2, 3$ を用いた.

となる。

この手法によって作成された辞書は、MWP, MWN がそれぞれ約 3 万個、約 10 万個の要素からなる。

3.2 複単語表現を考慮したことによる効果

実際に複単語表現を構成するそれぞれの単語でみるより、複単語表現をひとまとまりにして、事象の単位と捉えることで、複単語表現の意味をより正確に捉えた例を表 1 に示した。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 の値が 2.72, 2.20 と低い値となっているが、腰掛けるという意味の「take a seat」が結果のトークンである時、Causal Strength の値は 13.9 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 1 まとまりで捉えることで、その複単語表現の意味を捉えて、「wait」との因果関係をより正確に捉えることができている。

Table 1: 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	2.72
wait	seat	2.20
wait	take a seat	13.9
sun	cast	0.84
sun	shadow	25.7
sun	cast a shadow	81.0
think	come	3.63
think	come up with	5.27

4 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．

4.1 データ

Causal pair の抽出には ClueWeb12⁴ 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 億文抽出した．品詞タグ付け，lemmatization には Stanford Core NLP [12] を用いた．また，本手法で作成した辞書にある複単語表現が COPA の開発セットと評価セット全 1000 問の中に MWP, MWN それぞれ 514, 461 回登場した．COPA には多くの複単語表現が登場していることがわかる．

4.2 実験設定

COPA を解く際は causal pair を数え上げる時と同様に，前提文と候補文にある単語を全て lemmatize して，stop words，頻度上位 10 位以内の単語は除外し，名詞，動詞，形容詞，副詞でかつ WordNet に含まれる単語に限定した．作成した辞書中にある複単語表現が登場した場合，その複単語表現を 1 つの単語として扱った．結果を問われている問題では，前提文に含まれている単語を原因単語，候補文に含まれている単語を結果単語として扱い，それぞれの単語ペアのスコアを計算した．cause を問われている問題ではその逆である．各単語ペアのスコアの合計を正規化することで各候補文に対するスコアを計算し，スコアの大きい選択肢を選んだ．上記の前処理を施した 2 文 T_{pre}, T_{alt} のスコア計算を定式化すると以下のようなになる．

$$Score(T_{pre}, T_{alt}) = \frac{1}{|T_{pre}| \times |T_{alt}|} \sum_{i \in T_{pre}} \sum_{j \in T_{alt}} Score(i, j) \quad (10)$$

$|T_{pre}|, |T_{alt}|$ は文 T_{pre}, T_{alt} に含まれる単語の数である．Luo らは，スコア正規化のために $|T_{pre}| \times |T_{alt}|$ の代わりに $|T_{pre}| + |T_{alt}|$ を用いた．

Causal Strength の手法におけるハイパーパラメータ λ は開発セットでチューニングを行った．Causal Embedding の手法においては，ベクトルの次元は 1000 次元，負例サンプル時のサンプル数は 20，全データを 5 回繰り返し学習した．

⁴<http://lemurproject.org/clueweb12/index.php>

4.3 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 $|T_{pre}| \times |T_{alt}|$, $|T_{pre}| + |T_{alt}|$ であることを表す.

Table 2: COPA の実験結果

手法	コーパス	正答率 (%)
Causal Strength-add	Causal Net	70.2 %
Causal Strength-mul	ClueWeb12	69.9 %
Causal Strength-mul w/mwp	ClueWeb12	71.2 %
Causal Strength-mul w/mwn	ClueWeb12	68.5 %
Causal Strength-mul w/mwp, mwn	ClueWeb12	70.4 %
Causal embedding-mul	ClueWeb12	68.4 %
Causal embedding-mul w/mwp	ClueWeb12	67.6 %
Causal embedding-mul w/mwn	ClueWeb12	65.7 %
Causal embedding-mul w/mwp, mwn	ClueWeb12	63.1 %

Causal Strength 手法において, MWP を考慮したモデルが最も良い性能に達し, これまでの COPA の最高性能を 1 % 上回った. Causal Strength 手法においては, 複単語表現 MWP を考慮することが有効であることがわかった. 一方で, Causal Strength 手法と Causal Embedding 手法どちらにおいても, MWN を考慮したモデルは複単語表現を考慮しない時より性能が低下した. このことから MWN を考慮することは性能の低下につながるということがわかった. また, Causal Embedding 手法においては複単語表現を考慮すると, 複単語表現を考慮しない時より一貫して性能が低下した. このことから, 手法によっては複単語表現を考慮しても性能向上に繋がらないことがあるということがわかった.

ここで, 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人手で同定し, 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. その結果, 最大 71.0 %⁵ の正答率であった. この

⁵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.

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、我々の手法を用いたモデルは複単語表現を用いることでの性能改善の上限に達していると考えられる。

COPA をとく際にどの品詞が因果関係の推定において重要であるかを明らかにするために、COPA を解く際に、特定の 1 品詞の単語を除外する実験も行った。結果を表 3 に示した。ただしこの時、複単語表現は用いていない。

Table 3: 特定品詞を除外して COPA をといた時の正答率

	正答率 (%)
CS (名詞のみ除外)	58.0 %
CS (動詞のみ除外)	52.9 %
CS (形容詞のみ除外)	64.9 %
CS (副詞のみ除外)	70.1 %

動詞を除外した時、最も悪い性能となったことから、COPA をとく際に、動詞の因果関係がもっとも重要な手がかりとなっていることがわかった。

4.4 分析

はじめに、うまくいった例を紹介する。以下のような問題が存在する。

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

What is the cause of this sentence ?

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 1 単語とみなし、正解することができている。実際に、Causal Strength の

値をみると (bedtime_c, shut_e) の Causal Strength の値は 1.7 であるが, (bedtime_c, shut off_e) の値は 12.3 であり, 他の causal pair 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。

ただし, 複単語表現を取り入れることで, これまで正解できていたものが不正解になってしまう例もある。

前提文: I clumsily *bumped into* the stranger.

What is the effect of this sentence ?

候補文 1: : I *ran away*.

候補文 2: : I apologized to him.

候補文 1 中の run away を複単語表現として捉えたとき, (stranger_c, run away_e) の Causal Strength が高い値をとってしまい, 誤った選択肢である候補文 1 を選択してしまった。

また, これは複単語表現を用いる手法と用いない手法のどちらにも言えることだが, 「man」や「boy」など人を表す単語を含む causal pair のスコアが相対的に高くなっている場合が多いことが分かった。「man」, 「woman」, 「boy」, 「girl」が登場する数を数え上げたところ, 1000 問中 326 問, 約 3 割の問題に登場していることが分かった。そこで「man」, 「woman」, 「boy」, 「girl」を Causal Strength 計算の対象から除外して COPA を解き直してみると, 71.8% の正答率となり, 性能が向上した。このことから「man」など人を表す単語は COPA をとく際には重要ではない単語であることが分かった。

5 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辞書を作成し，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．

謝辞

本研究を進めるにあたり，ご指導を頂いた乾健太郎教授，岡崎直観教授に感謝いたします。また，実験や論文執筆にあたって直接指導して下さいました井之上直也助教，高瀬翔氏に心より感謝いたします。最後に，日常の議論を通じて多くのご助言をくださった乾・岡崎研究室の皆様感謝いたします。

References

- [1] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.
- [2] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797. Citeseer, 2008.
- [3] Zhiyi Luo, Yuchen Sha, Kenny Q. Zhu, Seung won Hwang, and Zhongyuan Wang. Commonsense causal reasoning between short texts. In *KR*, 2016.
- [4] Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. Creating causal embeddings for question answering with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 138–148. Association for Computational Linguistics, 2016.
- [5] Mark Granroth-Wilding and Stephen Clark. What happens next? event prediction using a compositional neural network model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16*, pages 2727–2733. AAAI Press, 2016.
- [6] Roger C Schank and Robert P Abelson. *Scripts, plans, and knowledge*. Yale University New Haven, CT, 1975.
- [7] Karl Pichotta and Raymond J. Mooney. Using sentence-level lstm language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, pages 279–289, Berlin, Germany, 2016.
- [8] Ashutosh Modi and Ivan Titov. Inducing neural models of script knowledge. In *CoNLL*, volume 14, pages 49–57, 2014.
- [9] Karl Pichotta and Raymond J Mooney. Learning statistical scripts with lstm recurrent neural networks. In *AAAI*, pages 2800–2806, 2016.

- [10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [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*, pages 3111–3119, 2013.
- [12] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

付録

Table 4: 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		
A lead to B	A leads to B	A led to B
A leading to B	A give rise to B	A gave rise to B
A given rise to B	A giving rise to B	A induce B
A inducing B	A induces B	A induced B
A cause B	A causing B	A causes B
A caused B	B caused by A	A bring on B
A brought B	A bringing on B	A brings on B
B result from A	B resulting from A	B results from A
B resulted from A	the reason(s) for/of B	BE A
DET effect of A	BE B	A BE DET reason(s) of/for B
inter-sentence		
If A, then B	If A, B	B, because A
B because A	B because of A	Because A, B
A, thus B	A, therefore B	B, A as a consequence
Inasmuch as A, B	B, inasmuch as A	In consequence of A, B
B due to A	Due to A, B	B in consequence of A
B owing to A	B as a result of A	As a consequence of A, B
A and hence B	Owing to A, B	B as a consequence of A
A, hence B	A, consequently B	A and consequently B
A, for this reason alone , B		