Abstract

Causal relations are essential knowledge for interpreting discourse structure of text. This paper presents a method that extracts causal relations of lexical patterns in the form of quasi Horn clauses: for example, “ A acquires B ” AND “ B is located in X ” → “ the acquisition gives A operations in X ”. The input of the method is a relation tuple consisting of two entities and a verb, e.g., (A, acquire, B). The method finds coreferencial expressions of the given relation that mention the same event. We use nominal forms of verbs included in FrameNet for finding the coreference expression. If a nominal form of a relation occurs as the subject in the dependency tree of a sentence, the sentence is likely to describe what is caused by the event. Then the method uses several NLP techniques (part-of-speech tagging, coreference resolution, dependency parsing and named entity recognition) in order to build a lexical pattern containing the entities (A and B). However, such rules are so specific that computers cannot reuse the knowledge of causality relation: for example, “ the acquisition gives A operations in Nevada ”. Therefore, the proposed method generalize the rules by introducing a variable X and estimating the relation between X and A or between X and B. For evaluation, we asked human annotators to judge the correctness of the causal-relation rules. The result shows that the proposed method precisely extracts the causal-relation rules.

目 次

表 目 次

図 目 次

第1章　はじめに

1.1　序論

近年，計算機とネットワークの速度・容量の益々の発展により，とても人間の手では整理できないような大量の文書を対象として検索が行えるようになった．それにより，従来では考えられなかったような，人間と同じように質問を理解し，しかも人間よりも速くその答えを導き出すことのできるようなシステムが登場し，衆目を集めた．例えば，IBM が開発した Watson(図 1.1) システムは，いわゆるクイズ大会で人間に挑戦し勝利したことで，昨年話題になった．

そのようなシステムでは，情報源の情報をいかに構造化し，応用的な検索ができるようにするかが重要である．そのためには計算機に自然言語の構造や意味関係を理解させる必要がある．自然言語理解のためには，文書中から「誰が何をした」かを認識する必要があり，さらにその結果何が起こったか，という因果関係を認識し，抽出することが重要である．本研究では，そのような自然言語処理からの因果関係知識の獲得を目指す．

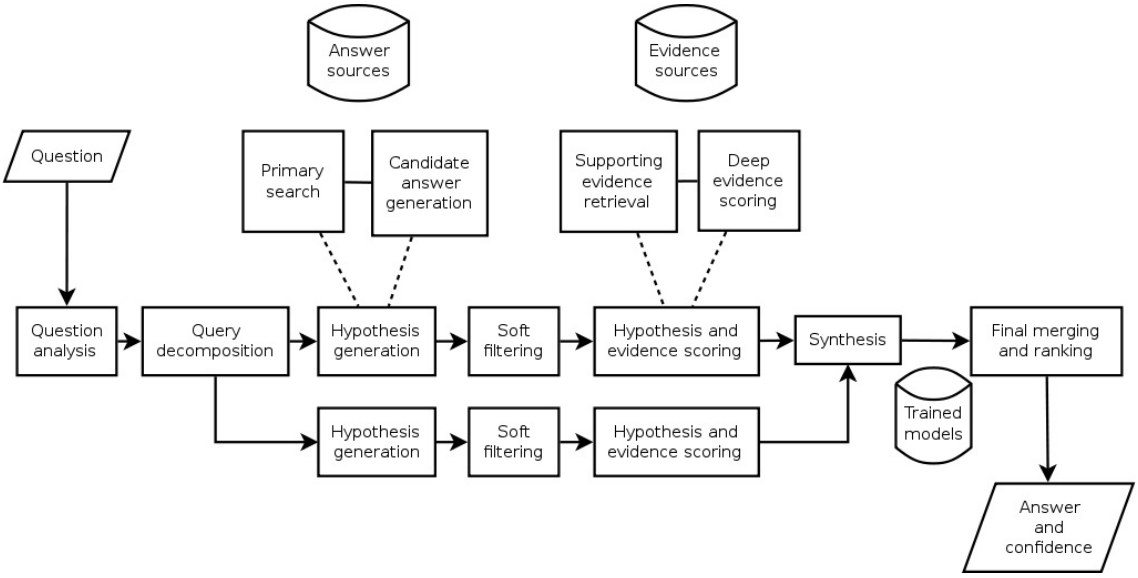

図 1.1: ワトソンの仕組み

1.2 本研究の目標

本研究の最終的な目標は，以下のような知識 (ルール) を抽出することである．

(A, buy, B) AND (B, has office in, X)
⇒ The acquisition gives A operations in X

このルールが意味するのは，A が B を買収し，かつ B が X に拠点を所持しているなら，その結果として A は X での事業展開を行うことができる，という意味である．本論文では,「A が B を買収する (“A buys B”)」のような,「誰」が「何」を「した」という事象のことを，**イベント**と呼ぶ．そして，複数のイベントが原因 (条件) で何らかの結果が起こるといったような，原因-結果の関係の事を**因果関係**と呼ぶ．

また，上記のルールが成立する理由に，結果節の “acquisition” が “buy” と同じイベントを指していることが挙げられる．本論文では，このように同じイベントを参照することを**イベント共参照**と呼び，特に参照に使われる表現を**イベント共参照表現**と呼ぶ．本研究では特に，このイベント共参照表現が記述する因果関係の抽出に着目する．

1.3 本研究の応用

因果関係の知識は，様々なアプリケーションに応用が可能である．例えば Watson のような質問応答 (Q&A) システムでは，因果関係は文書中に記述された知識を構造化するのに役に立つ．最も簡単な例として,「X の原因は何か」という質問に対して，X と因果関係にある知識 Y が分かれば，容易に答えることができる．

また，因果関係の知識は情報の整理，検索にも役に立つ．例えば大企業の中には，未整理のままの膨大な量のテキストを抱えている所もある．特にサポートセンターでの対応のログ等のデータであれば，そのデータ整理に因果関係の知識が役に立つ．すなわち，例えば「トラブル (原因)」→「対応 (結果)」を中心として情報を整理することができれば，トラブルに対して，どのような対処をすればよいのか，という情報の検索が容易に行える．

自然言語処理の学術的な立場からも，因果関係の知識は役に立つ．自然言語処理の中で難しい問題の 1 つに，代名詞等を解決する共参照解決がある．例えば，以下の例を考える．

A buys B. The acquisition gives the company operations in X.

この例の中で，“the company” が A と B のどちらを指しているのか，構文解析などからは明確には分からない．ここで，1.2 節で抽出した知識を事前に持っているとすると，“the company” が A を指すことが分かり．共参照解決を行うことができる．

他にも，例えば “acquisition” が与えられた時に,「誰」が「何」を買収したのかを知りたい場合がある．このような場合も，因果関係のような語彙間の関係を利用することで解決を行うことができる [1][15].

また，因果関係は Textual Entailment(第 2 章で説明) が成立するための条件の 1 つとして含まれる [21]．そのような研究分野に対しても，本研究は応用することが期待できる．

1.4 本研究の貢献

本研究の貢献は，以下の通りである．

- 本研究ではイベント共参照関係を利用した因果関係の抽出を行う．従来の研究では，イベント間の因果関係は抽出することができていたが，より直接的に「イベントが何を引き起こした」という知識には注目することができていなかった．本研究ではイベント共参照関係を利用し，より深い意味解析を行うことで，従来注目することのできていなかった因果関係の知識を獲得できる．

- これまでの研究では関係の間の関係というものについて，あまり議論はされてこなかった．本研究では入力として与えた関係と因果関係にある事象を，変数を含む語彙パターンの形で出力する．そして，そのパターンが成立するために必要な関係を外部リソースを用いて検索する．これにより，複数の関係から 1 つの事象が導かれるというルールが出力できる．すなわち，関係の間に成立しうる関係の中の 1 つであると考えられる因果関係を，1 つの語彙パターンの形で表現し，出力できる．

- 本研究において，イベント共参照表現として動詞の名詞化表現を利用している．自然言語処理において照応の解決問題や，共参照の解決問題は非常に難しいタスクであることが周知されている．本研究ではイベント共参照表現を利用して知識の獲得を行うが，評価実験においてその知識獲得の正確さを調査し，原因を分析する．それにより，イベントの共参照表現としての名詞化表現の妥当性を検証することができる．

- 本研究において，入力から変数を含む語彙パターンの獲得までは，品詞タグ付け，構文解析，固有表現抽出，共参照解決等の既存の自然言語解析の上に成り立つ基礎的なルールによって，教師無しのアルゴリズムによって処理が行われる．したがって，共参照表現を用いた知識獲得についての，基礎的なアプローチを示す．

1.5　本論文の構成

本論文の構成は，以下の通りである．

本章では研究の大まかな背景について説明し，本研究がどのように今後の発展的な研究に貢献するかについて述べた．第 2 章では本論文の背景を詳細に説明する．その中で関連研究の流れについて言及し，関連研究の問題点から，本研究の立ち位置について述べる．第 3 章では提案手法について，システムが入力を受け取ってから知識を出力するまで，その詳細を説明する．第 4 章では，提案手法を評価するために行ったいくつかの実験について，各実験の設定について説明し，その結果を示す．第 5 章では，評価実験の結果によって分かった提案手法の性能の分析と，問題点とその解決策について示す．また，提案手法の各手順について，より性能の良いシステムを構築するために改良可能な点について述べ，今後の発展的な研究への指針を示す．第 6 章では，結論と今後の展望を述べ，本論文の結びとする．

第2章　研究の背景

本節では，本研究と関連する研究を紹介し，本研究の位置づけを説明する．

2.1　因果関係の獲得

自然言語理解，情報抽出の研究において，因果関係は重要な研究分野の 1 つと見なされ，因果関係の自動獲得について多数の研究が行われてきた．

2.1.1　語彙パターンを利用した因果関係の抽出

因果関係を自動的に抽出する手法の中で最も単純なものは,「X が Y を引き起こした」のような語彙パターンを利用する手法である．Girju の研究では，質問応答システムへの応用のため，例えば，

mosquitoes cause malaria

というテキストの中の “cause” に注目することで，“mosquitoes” と “malaria” の因果関係を抽出している [16]．この研究で因果関係抽出の対象となっているのは名詞である．

そのような研究で最新のものに，Saeger らの研究がある [27]．この研究は日本語を対象としており，因果関係にある名詞の中で long-tail に相当する部分に注目している．

2.1.2　時系列データを利用した因果関係の抽出

因果関係とは，原因となる事象と，その結果として起こる事象の間に成立する関係であるため，事象の生じた時間と密接な関係がある．Sun らはこの点に注目して，検索エンジンのクエリのログから因果関係の抽出を試みた [29]．特に任意のクエリの検索回数に注目し，検索回数が急激に上昇した日時に何らかのイベントが生じたと見なし，そのようなイベント間の関連性を調べることで因果関係の抽出を行っている．

2.1.3 動詞間の因果関係の抽出

これまで紹介してきた研究はいずれも，名詞間の因果関係の抽出が中心であった．Beamer らは，脚本を利用して動詞間の因果関係を認識する研究を行った [2]．この研究では連続的に出現する動詞や，ごく近い距離で出現する動詞のペアに対する因果関係の認識を行ったが，認識できる因果関係はそのような連続的に出現する動詞のペアに限定されていた．

2.1.4 イベント間の因果関係の抽出

動詞間での因果関係の抽出の研究が行われると，今度は動詞の主語や目的語を含んだ「イベント」単位での因果関係の抽出が行われるようになった．Riaz らはイベントを表現する，ある程度まとまった量の文クラスタ間の因果関係を求める研究を行った [26]．この研究では入力テキストの文章を，動詞が “head” となったクラスタに分類し，1 つのクラスタが 1 つのイベントを記述するものとして，クラスタ間の因果関係を求めることで，イベント間の因果関係の抽出を行った．

この研究の発展的な位置づけにある最新の研究として，Do らの研究 [8] がある．この研究では文書内から，動詞が表現するイベントと，いくつかのルールを利用して作成した名詞化表現が記述するイベントを全て抽出し，それぞれのイベント間に因果関係が存在するかどうかを判定することで，因果関係にあるイベントのペアを抽出している．

本研究とは最終的に獲得したい知識も異なるが，特に名詞化表現の扱いが異なる．この研究では名詞化表現は動詞と同等のイベントを記述するものとして扱い，本研究のように動詞を受けるものとしては扱っていない．すなわち，例えば “A’s attack of B destroyed ...” のような表現が出現したとき，本研究では “destroy” 以降に特に注目するが，Do らの研究では “A’s attack of B” を “A attack B” というイベントとして抽出し，同様にして抽出された他のイベントとの因果関係を判定している．

2.2 Textual Entailment

本研究の応用的な位置づけにある研究分野として，Textual Entailment というものがある．ここではその定義と，いくつかその研究を紹介する．

2.2.1 TAC

Textual Entailment を扱う有名なワークショップに，Textual Entailment Challenge(TAC) がある．このワークショップは Textual Entailment の認識 (Recognizing Textual Entailment; RTE) の精度を競うものである．TAC が扱うタスクは 2 つあり，メインタスクでは，T と H という 2 つの文が与えられた時に，T が H を entail するかを判定する．TAC での Textual Entailment の定義は「人間が T という文から，H という文が真であることを推測できれば，T が H を entail する」となっている [7]．例えば，T，H のペアとして以下のような文が与えられる．

> H：South Ossetia is a separatist territory of Georgia.
> T：At least 17 Georgian soldiers were killed this summer in clashes with South Ossetian forces, raising tensions to fever pitch in the rebel territory, which like Abkhazia declared independence following a civil war in the early 1990s.

この例では，H が T から推測できるので，T が H を entail する，と判定される．メインタスクでは，entailment の定義に該当するものとしないもの，すなわち positive なものと negative なものに対応する T と H の文のペアが学習用に与えられ，参加者はその学習用データから知識を学習し，評価用のデータセットで精度を測る．

TAC のタスクは我々人間にも判断が難しい問題を扱うため，文の意味的なレベルに踏み込んだ研究は少なく，RTE システムのほとんどが品詞や係り受け関係など，文の構造レベルでの学習・評価を行っている[1]．

次に，Textual Entailment に関連する研究をいくつか紹介する．

2.2.2 動詞の Entailment グラフの作成

Berant らは，特に上位下位関係，及び言い換え関係にある動詞間での Entailment ルールの抽出を行うために，Entailment グラフ (Entailment 関係にある動詞を結んだグラフ)

[1]最近の RTE ワークショップでは，単純な単語の一致度を利用したシステムの方が良い性能を得ている．

上での線形計画法による最適化問題を解くことで，どの動詞間に Entailment ルールが存在するかを求めるアルゴリズムを提案した [3]．具体的には，まず動詞の上位下位関係が分かる WordNet[14] を利用して positive データと negative データを作成することにより，Entailment ルールの分類器を学習する．次に，Entailment ルールを抽出したい動詞の集合が与えられた時，分類器を元にしたルールの存在の可否と，動詞の 3 項間には Entailment のループが存在してはいけない，などのいくつかの束縛条件の下，与えられた動詞間で大域的な最適化問題を定義して解くことにより，精度の高い Entailment ルールの抽出を実現している．

Berant らは更に，動詞の主語と目的語になるエンティティの種類を限定した形での Entailment グラフを抽出する手法も提案している [4].

2.2.3 Web からのルール抽出

広く Web を対象として Entailment ルールの抽出を行った研究として，Schoenmackers らの研究 [28] がある．この研究では 1 次ホーン節を Web から教師無しで学習することを目的としている．この研究ではまず，関係抽出の研究で広く用いられている手法 [17] を用いて City などのクラス名と，そこに属する New York などのインスタンスを抽出する．次に，抽出したインスタンス間に存在する関係を抽出し，インスタンスのペアとその間の関係，という 3 つ組を得る．次に 3 つ組間に Entailment ルールが存在するかどうかを評価する．評価の中心になるのは，$p(C|A) >> p(C)$ という式 [30] である．C, A はそれぞれ 3 つ組のことであるが，この式が成り立てば，A は C と統計的に関係があるとみなされ，$A \rightarrow C$ のルールが生成される．この研究では，さらに $A \wedge B \rightarrow C$ となるような B についても帰納論理プログラミングの知識を元に抽出している．

2.2.4 日本語を対象とした研究

大友ら [31] は，日本語のテキストを対象として，述語項構造における項と用言の共起情報と節間関係の分布を用いて事態間関係の獲得を行った．この研究では例えば「X を焼く→ X が焦げる」といったような，時間経過や因果関係にある事態間関係を抽出している．そのために，まず順接や理由など，特定の係り受け関係にある述語項構造ペアを抽出・汎化し，述語項構造の種類により関係を分類する．例えば述語項構造のペアが (行為，出来事) であれば，両者は因果関係か時間経過の関係にある，と分類する．その後，述語項構造ペアの共起度を計算し，妥当なものだけを知識として獲得する．この研究は日本語の Web コーパスから幅広い事態間関係の抽出を 70%前後の精度で行うことができている．

2.3 関係抽出

本研究は関係抽出の応用的な位置にある．関係抽出とは，人名や企業名など，特定のエンティティ同士の関係を文章から抽出することである．文章としては，ニュース記事のようないわゆる「きれいな」のコーパスが利用されるだけではなく，近年の検索エンジンの著しい発展に合わせて，Web 検索エンジンの出力結果 (スニペットなど) も多々利用されている．ここでは，このような関係抽出についての関連研究をいくつか挙げる．

2.3.1 ブートストラップ

関係抽出におけるブートストラップとは，関係を記述した少数のデータを利用して，反復的な処理により多数の関係を取得する手法のことである [17]．具体的には，この手法ではまずシードとなる単語ペアを与える．それからコーパスを検索し，2 つの単語の間など，それらの周囲に現れる単語パターンを探し，抽出する．そして，今度は抽出した単語パターンが現れる箇所をコーパスから探し，シードとして与えた単語ペアと同じような関係にあると推定される単語ペアを新しく取得する．その後，取得した単語ペアを新しいシードとし，またそれらの周囲にある単語パターンを探す．この手順を繰り返すことで少数のデータから多数の関係を抽出することができる．Hearst によって提案されたこの手法は，当初いくつかの問題を抱えていた．しかし，KnowItNow[23] では正解率を，Espresso[25] では再現率を高める工夫がなされ，また DIPLE[5] や Snowball[10] では抽出パターン生成の自動化が行われ，更に Pasca らの手法 [22] や DIRT システム [20] では抽出パターンの抽象化が行われるなど，当初抱えていた問題を解決するような研究が多数なされている．

2.3.2 Open IE

OpenIE とは，入力として文章を与えたとき特定の分野に限定せず，そこに含まれる関係を全て抽出することを可能にするような枠組みのことを指す [9][11][24]．OpenIE は，Text Runner[2]というシステムが公開されており，このシステムは単語を入力すると，その単語が持つ関係をランク付きで出力する．また，Stat Snowball というシステムも存在しており [19]，人手で特徴量を決めなければならない Text Runner に対して，統計的な処理によって自動的に単語パターンを抽出することができる．

[2]http://www.cs.washington.edu/research/textrunner/

第3章　因果関係を抽出するための手法

本研究での最終的な出力は，単数あるいは複数のイベント (関係) が成立するとき，それによって導かれる知識 (そのイベントが原因で起こりうることなど) である．本章では，それを実現する提案手法について，各手順ごとにその詳細を説明する．

3.1　システム概要

本研究では入力として，最初に 1 つの関係を表す動詞を与える．その動詞から導かれる知識を探すために，その動詞が記述するイベントを参照する表現として，名詞化表現を用いる．本研究では，そのようなイベントの共参照表現が出現する文を深く解析することで，入力動詞が表現するイベント (関係) によって導かれる知識をパターンとして獲得する．更に，その知識が成立するために必要な別の関係を求めることで，複数の関係から導かれる知識を獲得する．提案システムは以下の手順で処理を行う．

1. 入力 (3.2 節)
   提案システムは，入力として因果関係を求めたい 1 つの動詞を受け取る．入力する動詞は人手で与えられる．

2. 名詞化表現の獲得 (3.3 節)
   本研究では，関係によって導かれる知識を獲得するためのイベント共参照表現として，動詞の名詞化表現を利用する．動詞の名詞化表現は，FrameNet(3.3.3 節で詳しく説明する) を利用して獲得する．

3. 文書の検索 (3.4 節)
   入力された動詞と FrameNet から獲得した名詞化表現を利用して，コーパスに対してそれらが共起する文書を検索する．

4. 文書の解析 (3.5 節)
   それぞれの文書に対し，因果関係獲得のために必要な，品詞タグ付け，構文解析，固有表現抽出，共参照解決を行う．

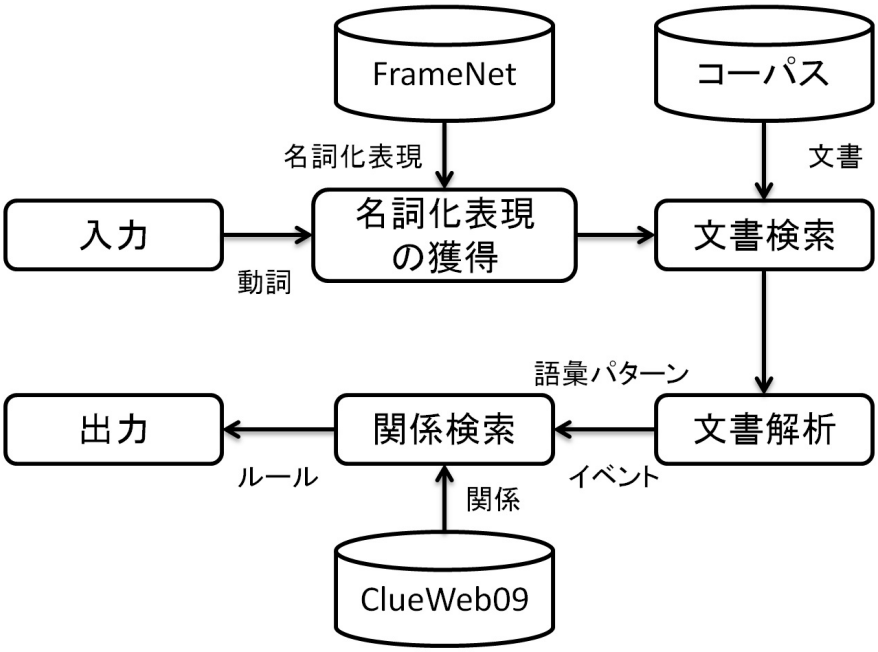

図 3.1: システム全体図

5. パターンの作成 (3.6 節)
   文書解析の結果を利用して，名詞化表現を含む構文木から因果関係の結果の部分を表現するパターンを作成する．

6. 関係の検索 (3.7 節)
   入力動詞が表現するイベントと，作成したパターンが表現するイベントの間にある関係を検索する．

7. 出力 (3.8 節)
   最終的に，入力動詞と検索された関係のペアと，それが原因で生じる結果を示すパターンがセットで出力される．

こうして，提案手法は入力として受け取った動詞に対し，因果関係にある知識を語彙パターンとして出力する．以後，各手順の詳細を説明する．

3.2 入力

提案システムは，入力として 1 つの動詞を受け取る．この動詞は，その動詞が原因となって生じるような因果関係を求める対象となるものである．ただし，以後の手順の中で用いている外部リソースや，アルゴリズムとの兼ね合い上，入力として与えることのできる動詞は，以下の条件を満たす動詞に限定する．

1. “A *verb* B” の形で記述することができる．ただし，A と B はそれぞれ，固有名詞 (エンティティ) でなければならない．

2. FrameNet(3.3.3 節で詳しく説明する) に掲載されており，同じ意味フレーム (これも 3.3.3 節で説明する)，または 1 つ上位の意味フレームの中に，名詞が 1 つ以上出現する．

そのような動詞に該当するものとして，例えば “A buys B” や，“A attacks B” などが挙げられる．本研究では，入力として与えられた動詞は 1 つのイベントを表現すると見なす．すなわち，例えば “A buys B” であれば,「A が B を買収した」というイベントを表現するものと見なす．以後の手順では，この「イベント」が引き起こす結果や事象を文書内から抽出することで，入力動詞と因果関係にある知識の獲得を目指す．

なお，入力される動詞に制限を求めることについて，その必要性と制限の緩和については．第 5 章で議論する．

3.3 名詞化表現の獲得

3.3.1 イベント共参照と因果関係

入力動詞が表現する特定のイベントによって引き起こされる結果や事象を，文書から抽出するための手法には，どのようなものが考えられるだろうか．考えられる最も自然な手法は，例えば「because」など，明示的に因果関係にある談話構造を示す表現を探すことである．すなわち，例えば “A buys B” によって引き起こされる事象を見つけたければ，

some event occurs **because** A buys B.

のような表現を探し，知識を抽出すれば良い．しかしながら，例えば企業の買収についてが良く書かれる，ニュース記事のようなコーパスにおいて，このように明示的に談話構造が書かれ，因果関係が抽出できることはほとんどない．このため，第 2 章でも述べたように，特殊な場合を除き，まず文書に存在する (あらゆる) イベントの抽出が先行し，

その後で各イベント間の関係を決定するという手法が一般的であった．本研究では因果関係の「原因」となるイベントを入力動詞の 1 つに絞り，その 1 つのイベントを参照する表現 (イベント共参照表現) の抽出をまず行う．抽出された各表現が含まれる文は当然，入力動詞が表現するイベントに関連する事柄の記述が含まれると考えられる．もしその表現が文 (あるいは句) の主語であれば，その文 (句) は「入力動詞が表現するイベント」が「何をした」という事柄を記述する．入力動詞が表現するイベントが主語になっているために，その「何をした」という事柄は，そのイベントが引き起こす事柄である可能性が高いと考えられる．本研究ではこの性質に注目して，因果関係を獲得する．

3.3.2 イベント共参照と名詞化表現

本研究では，関係によって導かれる知識を獲得するためのイベント共参照表現として，動詞の名詞化表現を利用する．今，入力されたのは動詞であるため，それを受け，参照する表現として考えられる中で最も自然なのは，その動詞の名詞化表現だからである．名詞化表現は例えば，入力動詞に単純に “-ing” を付加することで動名詞として簡単に得られる．しかしながら，例えば動詞 “buy” に対して “buying” が，イベント共参照表現として用いられることは考えにくい．すなわち，

A buys B. The **buying** causes ....

という表現は，やや不自然である．むしろ，

A buys B. The **acquisition** causes ....

という表現の方が自然であり，コーパス内での出現頻度も高いと考えられる[1]．“acquisition” は “buy” とは全く違う動詞の名詞化形であるが，我々人間は “acquisition” が「手に入れる」という意味において “buy” と近い意味を持っていることを知っているため，“A buys B” というイベントを参照していることが分かる．したがって，イベント共参照表現としての動詞の名詞化表現を求めるためには，“-ing” を付加するような文法的な単純なルールでは解決することはできず，動詞の意味のレベルで近い名詞を求めるための知識リソースが必要となる．そのようなリソースとして有名なものに WordNet[2][14] のような，単語の上位下位関係や，兄弟姉妹関係などを参照することのできるものもあるが，本研究では意味的に近い動詞・名詞の集合や，意味自体の上位下位関係が必要になるため，FrameNet を利用する．

[1]実際に後者に比べて前者のパターンの出現頻度は非常に少ない．(10 分の 1 程度;Google 検索のフレーズのヒット件数での調査)

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

表 3.1: FrameNet の意味フレームと，属する単語の例

フレーム	品詞	単語 (LU)
Event	動詞	go on, happen, occur, take place, transpire
	名詞	development, event
Getting	動詞	acquire, gain, get, obtain, procure, score, secure, win
	名詞	acquisition, procurement
Commerce_buy	動詞	buy, purchase
	名詞	purchase
Attack	動詞	ambush, assail, assault, attack, bomb, bombard, charge, hit, fall, infiltrate, invade, jump, lay, raid, set, storm, strike
	名詞	airstrike, ambush, assailant, assault, attack, attacker, fire, bombardment, bombing, charge, drive-by, incursion, invader, infiltration, offensive, onset, onslaught, raid, SAF, safire, small arms fire, strike
Apply_heat	動詞	bake, barbecue, blanch, boil, braise, broil, brown, char, coddle, cook, deep fry, fry, grill, microwave, parboil, plank, poach, roast, saute, scald, sear, simmer, singe, steam, steep, stew, toast
	名詞	なし

3.3.3 FrameNet

FrameNet[3][13] とは，国際計算機科学研究所 (ICSC) により運営されている，語彙データベースである．FrameNet はフレーム意味論 (frame semantics)[12] に基づいて構築されている．フレーム意味論の基本的な考え方は,「ほとんど単語の 1 つ 1 つの意味は，それぞれある 1 つの意味フレーム (semantic frame) という土台の上で理解される」というものである．意味フレームとは，単語の意味が表現するイベントや関係，エンティティの種類と，そのイベントや関係の参加者を記述したものである．例えば，料理 (cooking) という概念には，まず料理を行う人 (Cook)，そして料理される食べ物 (Food)，調理器具や入れ物 (Container)，熱源 (Heating_instrument) などが含まれる．FrameNet では，この概念は加熱 (Apply_heat) という意味フレーム (frame) として表現され，Cook や Food, Containter，Heating_instrument はフレームの要素 (frame elements) と呼ばれる．意味フ

[3]https://framenet.icsi.berkeley.edu/fndrupal/

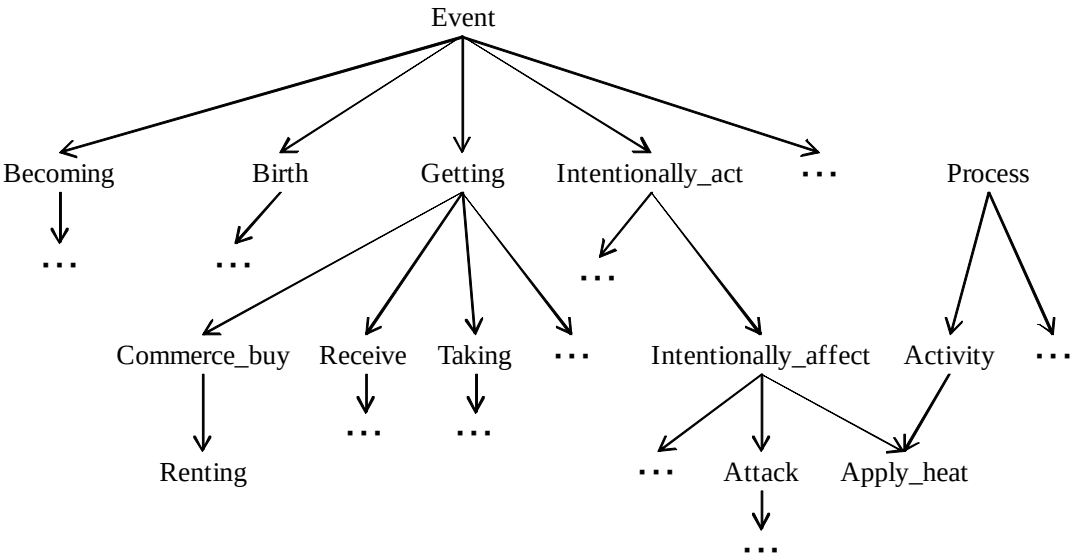

図 3.2: 意味フレームの上下関係の例

レーム「加熱」(Apply_heat) を引き起こすことのできる，“fry”, “bake”, “boil”, “brail” などの単語は Lexical Unit(LU) と呼ばれる．表 3.1 にこの意味フレームの具体例と，そこに属する単語 (LU) の例を示す.

FrameNet には意味フレームとそこに属する単語が記述されているだけでなく，意味フレーム間の相互関係も掲載されている．例えば，先に例として挙げた “Apply_heat” という意味フレームは，“Activity” というフレームと，“Intentionally affect” という意味フレームを継承しており，それらのフレームの下位の位置づけにある，ということが掲載されている．図 3.2 に，意味フレームの上位下位関係の例を示す.

ここまで説明してきたように，FrameNet には同じような意味を持つ単語のクラスタと，そのクラスタ間の相互関係が記述されているが，これらは全て人手で作成されたデータである．2012 年 2 月現在，FrameNet には 1135 個の意味フレームと，各フレームに属する 9796 個のフレーム構成要素が記載されている．意味フレーム間の関係は 1645 個，フレーム構成要素間の関係は 9439 個掲載されている．各意味フレームに属する単語 (LU) は 12387 個掲載されている.

3.3.4 名詞化表現の獲得

本研究では入力動詞が表現するイベントを受ける名詞化表現を，この FrameNet を利用して獲得する．名詞化表現の獲得は，以下の手順で行う．

1. 入力動詞が属する意味フレームを FrameNet から探す．例えば動詞 “buy” は意味フレーム “Commerce_buy” に属する．

2. そのフレームに属する名詞を抽出する．この時,「動作を行う人」を指す単語は除外する．具体的には，動詞+“er” や動詞+“or” で終わる名詞は除外する．例えば，動詞 “buy” に対して意味フレーム “Commerce_buy” から属する名詞 “purchase” を抽出するが，もし同じフレームに “buyer” のような動詞が所属していたら，その名詞は除外する．これは，そのような名詞はイベントを受ける表現とはならないことが明確だからである．

3. 名詞を抽出した意味フレームの上位会関係を調べて，親になるフレームから同様に名詞を抽出する．例えば “Commerce_buy” の親となるフレームは “Getting” であり，そこに属する名詞である “acquisition” や, “procurement” を抽出する．

入力動詞の属する意味フレームの親フレームからも名詞を抽出するのは，知識抽出の対象となるコーパスからより多くのイベント共参照関係を抽出するためである．あるフレームに対してその親フレームは，子フレームと意味的な包含関係にあると考えられる．そのため，例えば動詞 “buy” に対して “acquisition” をそのイベントを受ける名詞化表現として用いることができるように，入力動詞が属する意味フレームの親フレームに所属する名詞は，入力動詞を受ける名詞化表現となりうる．

入力動詞の属する意味フレームに，入力動詞と共に属する動詞は，入力動詞とその意味において類義語と見なすことができる (“buy” と “purchase” など)．本研究では同じ意味フレームに属する動詞と名詞は，イベントの共参照関係にあると見なすため，そのような FrameNet 内で類義語にある動詞も，入力動詞の代替表現として用いる．すなわち，名詞化表現の抽出と同時に，入力動詞と同じフレームに属する動詞も全て，与えた動詞と同じ意味を表現するものとして抽出する．

したがって，提案システムは入力として 1 つの動詞を受け取るが，FrameNet を用いて意味フレームという形で動詞の拡張を行うため，結局 1 つの意味フレームを対象として，その意味フレームと因果関係にある知識を抽出することになる．なお，多くの動詞は複数の意味を持ち，したがって複数の意味フレームに所属しているが，ここではその中で 1 つの意味フレームを対象として，動詞の拡張と，共参照表現としての名詞化表現の抽出を行うものとする．

3.4 文書の検索

次に，知識抽出に用いる文書を獲得する．FrameNet から獲得した動詞と名詞化表現を利用して，コーパスに対してそれらが共起する文書を検索する．この時，動詞，名詞化表現の中で，それぞれいずれか 1 つでも文書内に出現すれば，知識抽出対象とする．つまり，例えば動詞 “buy” と名詞 “acquisition” が共起する文書だけでなく，動詞 “purchase” と名詞 “acquisition” が共起した文書も獲得する．

本研究では用いるコーパスを，ある程度短いニュース記事程度の長さの文書が，大量に集積されたようなものを想定している．したがって，1 つの文書中に出現する動詞と名詞化表現のペアは，互いに共参照関係になる可能性が高いものとして扱う．

3.5 文書の解析

文書を獲得したら，それぞれの文書に対し，品詞・構文解析，固有表現抽出，共参照解決 (この場合の共参照解決は代名詞等の解決である) を行う．各処理は全て，Stanford CoreNLP[4]を用いる．

3.5.1 Stanford CoreNLP

Stanford CoreNLP とは，Stanford Natural Language Processing Group により開発されている，自然言語解析処理ツールである．本研究では，動詞と名詞化表現が共起する文書に対して，以下の解析を行う．

前処理

前処理として，文書を文単位で分割し，更に単語単位に分割する．特に本研究では，後の処理でエラーを起こしやすい，“—” で囲まれる表現等はあらかじめ除去しておく．

品詞タグ付け

品詞タグ付けでは，分割された単語に対して，品詞を判定してラベル付けを行う．例えば，出力として以下のようなものが得られる．

This/DT is/VBZ a/DT sample/NN sentence/NN

“DT” は determiner(限定詞)，“VBZ” は verb，present tense，3rd person singular(動詞の三人称現在形) を表すといったような，品詞のタグ付けが行われる．こ

[4]http://nlp.stanford.edu/software/corenlp.shtml

表 3.2: 固有表現のタグ一覧

タグ	説明	例
Time	時刻を表現する	12:00, 10 pm
Location	場所を表現する	United States, Japan
Organization	会社名など組織名を表現する	IBM, FIFA
Person	人名を表現する	Michael Jackson, Obama
Money	金額を表現する	$100,000, 100 yen
Percent	割合を表現する	97.79%, 50 percent
Date	日付を表現する	1999, 4 April
Number	数字を表現する	1, ten

の品詞タグは英語のコーパスとして有名な Brown Corpus の Tag セットを利用している[5].

固有表現抽出

固有表現抽出では，名詞の中で，一般名詞でないものに対して，その名詞の種類を示すタグ付けを行う．タグ付けに使われる固有表現の種類の一覧を，表 3.2 に示す．例えば，固有表現抽出によって以下のような結果を得る．

> Google/ORGANIZATION was/O founded/O by/O Larry/PERSON Page/PERSON and/O Sergey/PERSON Brin/PERSON

構文解析

構文解析では品詞タグ付けの結果を元に，それぞれの単語の係り受け関係を求める．例えば，

> Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas

という文を入力すると，図 3.3 のような構文木を結果として得る．図 3.3 の根 (root) は “submitted” であり，“Bills” が “submitted” に対し，nsubjpass(受動節の名詞主語) として係る．このように，構文解析の結果，どの単語がその文の主語であり，動作対象が目的語あるか，という情報を知ることができる．

[5]http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html

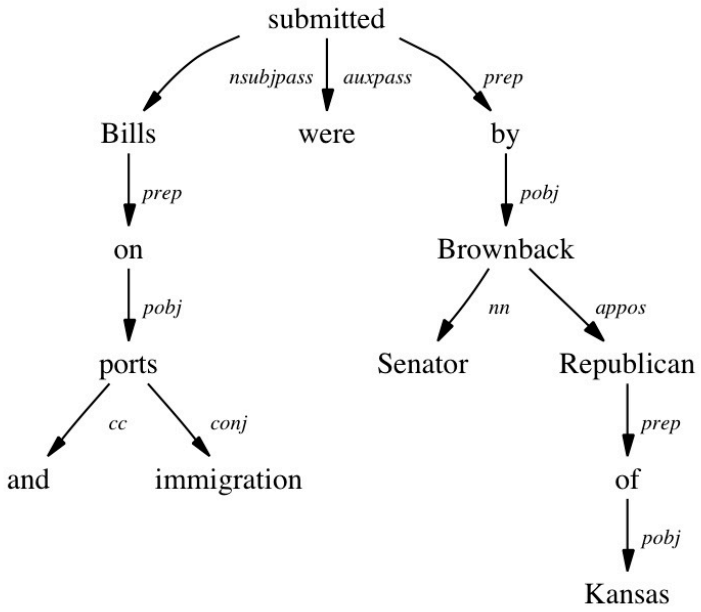

図 3.3: 構文解析の例

共参照解決

共参照解決では，文書中に出てくる固有名詞や，代名詞の中から，どれとどれが同じ人・物を指しているのかを解決する．例えば，

I bought a cake. I ate it.

という文書が与えられると，1 文目の “I” と 2 文目の “I” は同一人物で，2 文目の “it” は 1 文目の “cake” を指す，という情報が得られる．

以上，4 つの解析処理を行うが，後者になるほど難易度が高い．特に共参照解決は自然言語処理において難しい問題として知られており，結果に誤りが生じることも往々にしてある．本研究ではこの解析処理の結果を正しいとして以降の処理を行うが，解析の誤りによって以降の処理に誤りが生じることもあるため，簡単なルールによるフィルタリングを行うことで，誤りを避ける処理を行う．

解析後，その結果は全て保持しておくが，特に共参照解決の結果により参照先の分かった代名詞などについては，その参照先で置き換えて以降の処理を行う．

3.6 パターンの作成

次に，構文解析を行った結果をもとに，入力動詞が表現するイベントが引き起こす結果・効果を表現する語彙パターンを作成する．

3.6.1 入力動詞の処理

最初に，入力動詞 (及び FrameNet で拡張した動詞; 以降「入力動詞」はそのような動詞も含むとする) が表現するイベントを抽出する．すなわち，入力動詞 “buy” に対し,「誰が」「何を」“buy” したのか，という知識の抽出を行う．

抽出のために，まず文書内で入力動詞を含む文を探す．構文解析の結果を調べ，動詞の主語と目的語になっている単語を抽出する．具体的には，動詞に表 3.3 に示す関係で係るものを主語，表 3.4 に示す関係で係るものを目的語として抽出する．

表 3.3: 主語として抽出する係り受け関係のルール

依存関係	説明	例
nsbj	名詞主語	**Clinton** defeated Dole
agent	受動態に “by” で係る補語	The man has been killed by the **police**
poss	所有	their offices

表 3.4: 目的語として抽出する係り受け関係のルール

依存関係	説明	例
dobj	直接目的語	She gave me a **raise**
sbjpass	受動態の主語	**Dole** was defeated by Clinton
iobj	間接目的語	She gave **me** a raise

この抽出作業の結果，(主語，入力動詞，目的語) という 3 つ組が抽出できる．本研究ではこの 3 つ組を，関係抽出における 1 つの関係と同一視する．このため，関係抽出における関係のように，3 つ組は「2 つのエンティティとその間の関係」となっていることが望ましい．したがって，固有表現抽出及び共参照解決の結果を参照して，主語と目的語が固有表現でない場合は，知識を抽出しない．3 つ組に未解決の代名詞や一般名詞が含まれる場合は雑音のような関係が多いため，この処理は以降の処理の精度を上げる点でも必要である．なお，ここでの固有表現は，表 3.2 の中で関係抽出におけるエンティティとなる，“Location”，“Organization”，“Person” にタグ付けされたものとする．

最終的に，例えば “A buys B”(A，B がそれぞれ別々のエンティティ) のような 1 つのイベント表現が獲得できる．なお，文書内でそのような知識が 1 つも抽出できない場合は，その文書を以後の処理から除外する．また，1 つの文書から複数の 3 つ組が抽出できる場合は，主語と目的語になるエンティティを全て保持しておく．すなわち，抽出される知識は “A *verb* B” と，A に入りうるエンティティのリスト，B に入りうるエンティティのリスト，ということになる．

3.6.2 語彙パターンの作成

次に，獲得した動詞によるイベントを受ける名詞化表現を含む文から，パターンを作成する．本研究では動詞が表現するイベント (関係) が導く知識を獲得することを目的とするため，名詞化表現が主語になるようなパターンを探す．すなわち，まず文書内で FrameNet から獲得した名詞のいずれかを含む文を探す．その文の構文解析の結果の係り受け木から，その名詞が主語になる木を抽出する．主語を抽出するルールは，動詞の場合と同じで表 3.4 のルールを用いる．

次に，探した木の中で，根 (root) となる動詞を探す．本研究では，名詞化表現が主語となったこの木が，入力動詞のイベントが引き起こす結果・効果を表現するものと見なす．したがって，root となる動詞を中心として，再利用可能な知識を表現するパターンを作成するため，以下の手順でその木の抽象化を行う．

1. root から見て，構文木上深さ 4 以下の単語は root への意味的な修飾関係としては遠く，パターンに不必要と見なし，削除する．

2. root から見て，深さ 3 の単語は root への意味的な修飾関係としては遠いが，木の抽象化を行う上でパターン上の置き換え可能な変数と見なし，それを表現する “X” などに置き換える．ここで，置き換えた “X” に元々入っていた単語は別途保持しておく．もしその単語が名詞句の一部 (すなわち，名詞修飾でその単語に係る単語がある) であれば，その名詞句全体を保持する．

3. 深さ 2，あるいは 1 の単語は root の動詞の直接的な主語・目的語であったり，その主語や目的語を直接修飾するため，パターンには必要であると見なし，そのまま木に残す．

4. 残った単語と “X” で置き換えた単語の中で，A や B(3.6.1 で抽出したイベントの主語，及び目的語) に入りうるエンティティと同一のものと見なすことができるものがあれば，変数 “A” や “B” で置き換える．本研究では，以下の条件に該当する場合，ある単語と 1 つの固有名詞が同一であるものと見なす．

表 3.5: 不必要であると見なす係り受け関係のルール

依存関係	説明
advcl	副詞節の修飾
advmod	副詞修飾
appos	同格
amod	形容詞修飾
complm	補語
dep	何らかの依存
mark	副詞節の説明
quantmod	量的修飾
rcmod	関係節修飾
xcomp	不定詞修飾
xsubj	形式的主語

- その単語が，固有名詞と全く等しい場合
- その単語が，固有名詞を構成する単語の一部を含む場合．例えば，固有名詞が “Google Inc.” である場合，“Google” は “Google Inc.” と同一視する．
- その単語が，固有名詞を構成する単語の頭文字からなる場合．例えば，“Hewlett-Packard” はその略称である “HP” と同一視する．

5. 残った単語と “X” で置き換えた単語の中で，固有表現抽出で数字や金額等を示す表現であることが分かっている場合には，“MONEY” などの特殊な変数で置き換える．置き換える対象は表 3.2 の中で，“Time”，“Money”，“Percent”，“Date”，“Number” である．残った単語の中で “Location”，“Organization”，“Person” にタグ付けされた固有表現で，上記の置き換えが行われなかった単語は，変数 “X” で置き換える．

6. 最後に，特定の修飾関係にある枝は削除する．これは，root との距離が近くても，例えば同格表現や副詞による修飾，“and” 表現などは語彙パターンには不必要であると考えられるからである．表 3.5 に不必要であると見なした修飾関係の一覧を示す．また，この木の主語である名詞化表現の修飾語も，不必要として除去する．

以上の手順で，変数を含む語彙パターンが獲得できる．図 3.4 に入力動詞が “buy”，名詞化表現が “acquisition” であったとき，

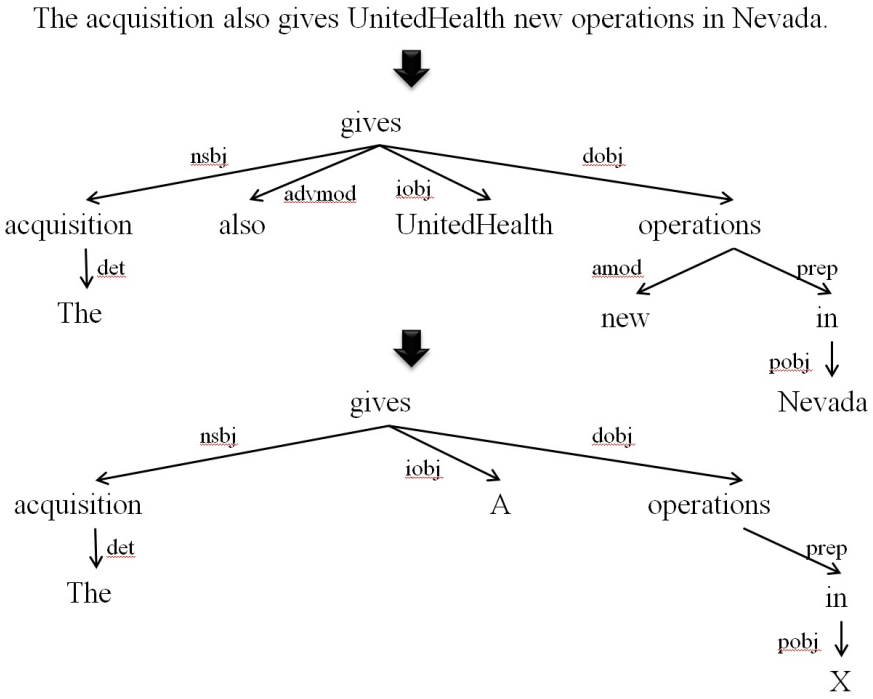

図 3.4: 構文木とその抽象化の例

UnitedHealth buys Pacificare.

The acquisition also gives UnitedHealth new operations in Nevada.

を構文解析し，語彙パターンを作成するときの様子を示す.

3.7 関係の検索

作成した語彙パターンには変数が入っているが，各変数の中にはどのような単語が入ってもよい，という訳ではない．変数 “A” や “B”，または “MONEY” のような変数は入る単語に対して制限を与えているが，“X” に関しては現段階では何の制限も与えていない．本研究では “X” についての制限を与えるために，“X” と “A” や “B” との関係を求める．関係を求めることで，例えば “A *verb* B” と “X *verb* B” が成り立つ時，作成した語彙パターンが成り立つ，という形での因果関係の知識を抽出することができる．

新たに “X” と “A” や “B” 関係探すのに最も精度の良いと考えられる手法は，今因果関係の抽出の対象となっている文書中から “X” に元々と入っていた名詞句と，“A” や “B” との関係を抽出する手法である．しかしながら，本研究では文書 1 つのサイズをニュース記事のようなある程度短いものと想定しており，語彙パターンと同一文書中から正確にそのような関係が抽出できることは少ないと考えられる．例えば，“B” に入りうるエンティティが会社名であり，“X” で置き換えた名詞が “B” の本社所在地であった場合，“B is headquotered in X” のような関係の抽出を本研究では目標とするが，ニュース記事の場合そのような関係は読者があらかじめ分かっている情報と想定していることが多いため，同じ記事の中には書かれにくい．したがって，本研究ではこの問題を解決するため，以下のような手法を採用する．

- ClueWeb09[6]から Reverb[7]によって抽出された大量の関係の中から，片方のエンティティが “X” に元々入っていた名詞句 (及びそれに近い名詞を含む) であり，もう片方のエンティティが A 及び B である関係を (複数) 探し，探すべき関係の候補とする．
- 事前に作成した学習データをもとに線形分類器を作成し，候補をスコア付けし，最も高いスコアを持つ関係を出力する．

3.7.1 Reverb

Reverb とは，英語の文章から 2 つのエンティティとその間の関係，という知識を自動的に抽出することのできるツールであり，第 2 章で紹介した TextRunner の内部で用いられている．本研究では関係の候補を探すために，リソースとして Reverb を ClueWeb09 に対して適応したものを用いる．ClueWeb09 は Web ページをクローリングし，大量に蓄積したコーパスである．このコーパス上で Reverb を動かし，抽出した関係のデータが公開されているため，本研究ではこれをリソースとして用いる．このリソースには，(entity1, *relation*, entity2) の形での約 1500 万の関係の知識が含まれる．

[6] http://lemurproject.org/clueweb09.php/
[7] http://reverb.cs.washington.edu/

3.7.2 検索

A 及び B と，保持しておいた各変数に元々入っていた名詞句を用いて，両者を entity1 あるいは entity2 として所持する関係を探す．ただし，ここで検索する対象は，“A” と “B” の中で語彙パターンに含まれないものを片方に持つ関係である．すなわち，例えば語彙パターンに “A” が含まれるなら，“B *verb* X” のような関係のみを検索する．もし両方とも含まない場合は，以下の処理は “A” と “B” 両方 (つまり，“A *verb* X” と “B *verb* X” の両方) に対して行う．

検索は原則として entity1 や entity2 と各名詞句が完全に一致するものを優先して行う[8]．もし関係が 1 つも見つからない場合は，各名詞句の一部などを用い，条件を緩めた形で検索する．それでも見つからない場合は，“A” や “B” を片方のエンティティに持つ関係を検索する．最終的に，検索結果の関係の Reverb のスコア[9]での上位 100 件[10]を候補として取得する．候補が 1 つも抽出できなかった語彙パターンは，以降の処理から除外する．

3.7.3 関係の選択

次に，取得した候補をスコア付けして，各 “X” と “A” や “B” との関係を表現するのに相応しい関係を 1 つ選択する．本研究ではあらかじめ学習しておいたサポートベクターマシーン (SVM) の分類器を用いて，スコア付けを行う．まず，学習データの作成について述べる．

学習データの作成

作成するデータは，ある語彙パターンとその中に含まれる “X” と “A” や “B” との関係として，何が相応しく，何が相応しくないのか，というデータである．今回は “acquire”(意味フレーム “Getting”) という 1 つの動詞を学習データの作成に用いた．これまで説明した手順に従って語彙パターンの作成を行い，候補となる関係の検索を行い，それぞれの “X” に対して 100 個の候補を得る．それぞれの候補に対し人間の評価者が，それが ‘X”

[8]本研究の検索エンジンは，MySQL で実装している．MySQL の全文検索にはクエリ拡張の機能があり，検索したクエリそのものを含まずとも，そのクエリと関連の深い単語が含まれる場合，そのレコードを検索結果に加えることができる．本研究ではこの機能を利用して検索を行っている．

[9]Reverb のスコアには 2 種類ある．1 つはその関係の抽出が正しい (その関係が関係としてきちんと成り立っている) かどうかを示すスコアで，もう 1 つはその関係のコーパス内での出現頻度である．ここでのスコアは前者を指す．

[10]この値は変更可能なパラメータである．本研究では処理速度の兼ね合い上，100 という数値を用いた．

と “A” や “B” との関係として相応しいかどうか，という positive/negative のラベル付けを行う．今回は 10 個のパターンに対してこのラベル付けを行った．

次に，作成した学習データをもとに，SVM の分類器を学習する．SVM の学習には特徴量を設定する必要があるが，今回は以下の特徴量を用いた．

1. エンティティ同士の類似度
   候補となる関係の entity1，entity2 の中で，“A” や “B” ではない方のエンティティ (つまり，“X” に元々入っていた名詞句と一致するものと見なしたエンティティ) と，“X” に元々入っていた名詞句との単語類似度．類似度の計算にはコサイン類似度

$$cosine(v_i, v_j) = \frac{v_i \cdot v_j}{|v_i| \cdot |v_j|}$$

   を用いる．この時のベクトルは単語の出現頻度ベクトルである．

2. パターンの類似度
   候補となる関係の *relation* と，語彙パターンを抽出した元のニュース記事との単語の出現類似度．類似度の計算には同様にコサイン類似度を用いる．

3. パターン同士の相関
   抽出した語彙パターンと，候補となる関係の *relation* の出現する文書の相関．相関の計算には Jaccard 係数

$$\text{Jaccard}(\text{V}, \text{U}) = \frac{|\text{V} \cap \text{U}|}{|\text{V} \cup \text{U}|}$$

   を用いる．V は語彙パターンが出現する文書 ID の集合で，U は *relation* が出現する文書 ID の集合である．ただし，語彙パターンには変数等が含まれ検索が困難であるため，文書への出現の検索はパターンを構成する単語の AND 検索で行う[11]．

4. エンティティ同士の相関
   候補となる関係の entity1，entity2 の中で，“A” や “B” ではない方のエンティティと，“X” に元々入っていた名詞句との相関．相関の計算は Jaccard 係数を用い，出現する文書の検索も語彙パターンの場合と同様．

5. 元々の名詞句と *relation* の相関
   “X” に元々入っていた名詞句と，*relation* の相関．相関の計算は Jaccard 係数を用い，出現する文書の検索も語彙パターンの場合と同様．

[11] 検索が困難であるだけでなく，フレーズ検索を行った場合は検索結果が低頻度の場合が多いことも問題となる．

6. 語彙パターンとエンティティの相関
   候補となる関係の entity1，entity2 の中で，“A” や “B” ではない方のエンティティと，語彙パターンの相関．相関の計算は Jaccard 係数を用い，出現する文書の検索も語彙パターンの場合と同様．

7. エンティティの文脈類似度
   候補となる関係の entity1，entity2 の中で，“A” や “B” ではない方のエンティティと，“X” に元々入っていた名詞句の用いられる文脈の類似度．まず，文書への出現の検索と同様の方法で，それぞれが出現する「文」の検索を行う．次に，検索された文に含まれる単語の出現頻度ベクトルを作成し，文脈ベクトルとする[12]．両者の文脈ベクトルのコサイン類似度を計算することで，文脈類似度を求める．

以上の特徴量の値を各関係の候補に対して計算し，positive/negative データと合わせて学習データとする．これを利用して SVM のモデルを学習する．モデル学習の際のパラメータ調整などは，第 4 章にて説明する．

関係の選択

学習したモデルを利用して，関係を 1 つ選ぶ．候補である全ての関係に対して，学習データ作成の際に利用した特徴量を全て計算し，SVM のモデルによってスコアを計算する．そのスコアの最も高い関係を，“X” と “A” あるいは “B” との関係を表すものとして最も相応しいとして，選択する．

[12]本研究では数百万単位の文書が含まれる巨大なコーパスを用いることを想定しているため，検索結果の全ての文・全ての単語を用いて文脈ベクトルを計算しようと思うと非常に時間がかかる．そこで本研究では，出現ベクトルの要素となれる単語を予め絞る．そのためにコーパス内のストップワード以外の全単語をその出現頻度順に並べ，上位 10000 個を抽出する．この 10000 個の単語のみを文脈ベクトルの要素とし，更に解析する文も 100 のみとする．本研究では，こうしてできた近似的な文脈ベクトルを代替として用いている．

3.8 出力

最終的な出力の前に，語彙パターンとして誤りである可能性の高い以下のパターンは除外する．

- 語彙パターンの品詞解析を行った場合，最後の単語が名詞でない場合．この時，何らかの原因で語彙パターンが修飾の途中で途切れていることが考えられるためである．
- 変数が連続する場合．例えば “give A X” のような語彙パターンの場合，変数が連続していても正しいと言えるが，本研究で採用したルールでは変数が連続する場合は意味のない語彙パターンであることが多いため，除外する．

以上の一連の処理によって，例えば “A buys B” と “B *is headquotered in* X” が成り立つ時，X を含むパターン (例えば “The acquisition give A an advantage in X”) が成り立つといったような，複数の関係を結び付ける知識を獲得することができる．これが，本研究の最終的な出力となる．

第4章　実験

本章では，提案手法に必要なパラメータ設定のための予備実験と，提案手法の評価実験の結果について述べる．以降の実験は全て，English Gigaword Corpus Third Edition(LDC2007T07; EGC) をコーパスとして用いている．まず，このコーパスについて説明する．

4.1　English Gigaword Corpus Third Edition

English Gigaword Corpus は，いくつかの新聞社について，英語のニュース記事を数年間分蓄積したコーパスである．ペンシルバニア大学の Linguistic Data Consortium(LDC) により作成されている．今回は，このコーパスの第 3 版を用いる[1]．表 4.1 に，第 3 版で蓄積された新聞社と，その記事の量を示す．

今回の実験ではこの中で，特に L.A. Times を実験に用いた[2]．

表 4.1: English Gigaword Corpus Third Edition の統計的データ

新聞社	期間 (月)	文書総数	トークン数 (1,000 個)
AFP 通信	98	1592309	466718
AP 通信	145	2272995	849435
CNA	96	85600	21657
L.A. Times	91	295224	192650
New York Times	149	1655279	1188494
新華社通信	143	1247039	249521

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2007T07

[2] 本来はコーパス内の全ての記事を用いて実験を行うべきであるが，今回の評価実験では人間による評価を行うものが多いため，文書の解析処理速度と評価者への負荷を考慮し，特に L.A. Times の 1 つを用いた．

表 4.2: 学習データの正負の割合

語彙パターン	“X” の総数	関係の総数	positive	negative
10	10	1064	80	984

4.2 SVM のパラメータ調整

3.7.3 節で説明したように，本研究では因果関係知識作成の中で必要な関係の選択のために，サポートベクターマシーン (SVM) によるスコア付けを行う．SVM では学習の際，学習器のパラメータを調整する必要がある．ここではそのパラメータ設定のための予備実験の手順と，結果を示す．

4.2.1 実験設定

3.7.3 節で説明したように，学習データとして，positive/negative のラベル付けと，7 次元の特徴ベクトルのペアが，語彙パターン 10 個それぞれについて 100 個ずつ，約 1000 個用意されている．表 4.2 に学習データの統計データを示す．

この学習データをもとにサポートベクターを学習する．今回の実験では，SVM の学習・分類を行えるツールとして，SVM Light[3]を用いた．今回，調整したパラメータは以下の 2 つである．

- c: 誤識別率とサポートベクターのマージンの比率
- j: positive/negative の識別エラーに対する重みづけの割合を決めるコスト係数 [18].

2 つのパラメータの調整を，以下の手順で行う．

1. 現在のパラメータでサポートベクターを学習する．
2. 学習したサポートベクターの分類性能 (Precision/Recall) を調べる．Precision，Recall の定義は以下の通りである．

$$\text{Precision} = \frac{\text{true positive}}{(\text{true positive}) + (\text{false positive})}$$

$$\text{Recall} = \frac{\text{true positive}}{(\text{true positive}) + (\text{false negative})}$$

3. パラメータを更新して再びサポートベクターを学習する．

[3] http://svmlight.joachims.org/

4. 分類性能 (F_1 値) の最も良いパラメータを最終的なパラメータとして決定する．F_1 値の定義は以下の通り．

$$F_1 = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

パラメータ調整は j，c の順で行う．つまり c を固定[4]したまま j を調整した後，その j を用いて c を調整する．パラメータの調整は j を 1 刻み，c を 0.1 刻みで変化させて行う．

なお，本研究では SVM のカーネル関数は用いなかった．

4.2.2 実験結果

図 4.1 に j を変化させた時の分類性能，図 4.2 に c を変化させた時の分類性能[5]を示す．

実験の結果，パラメータを j=6 で F_1=35.68 で最大値 (図 4.1) となった．j=6 を固定して c を変化させた所，c=1 で F_1=35.68 で最大値 (図 4.2) となった．

この実験の結果，j=6，c=1 と決定し，このパラメータを利用してサポートベクターを学習した．以降の実験では，ここで作成したサポートベクターを用いて関係の選択を行う．

[4]今回の実験では，c のデフォルトである 1

[5]c の値によるデータのばらつきが存在するが，これは学習器の収束アルゴリズムが原因と推測される．

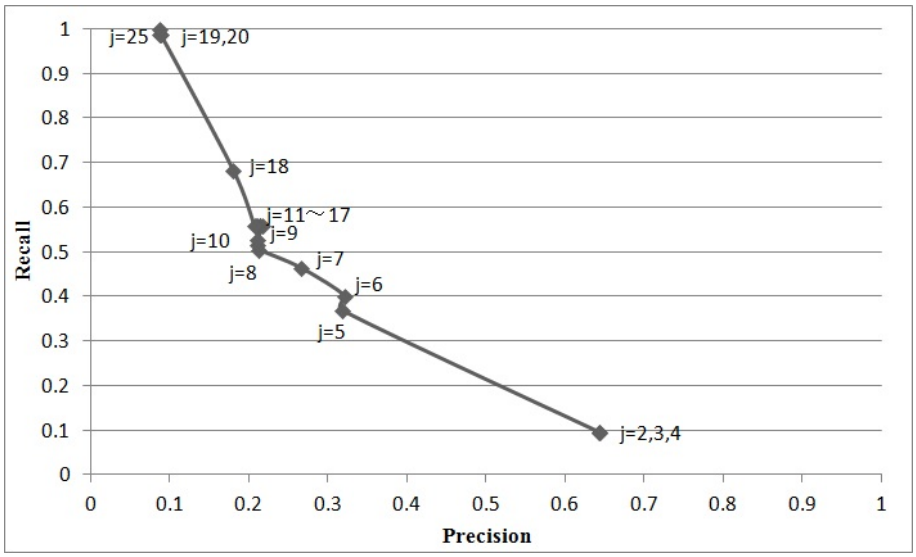

図 4.1: パラメータ j の変化と分類性能

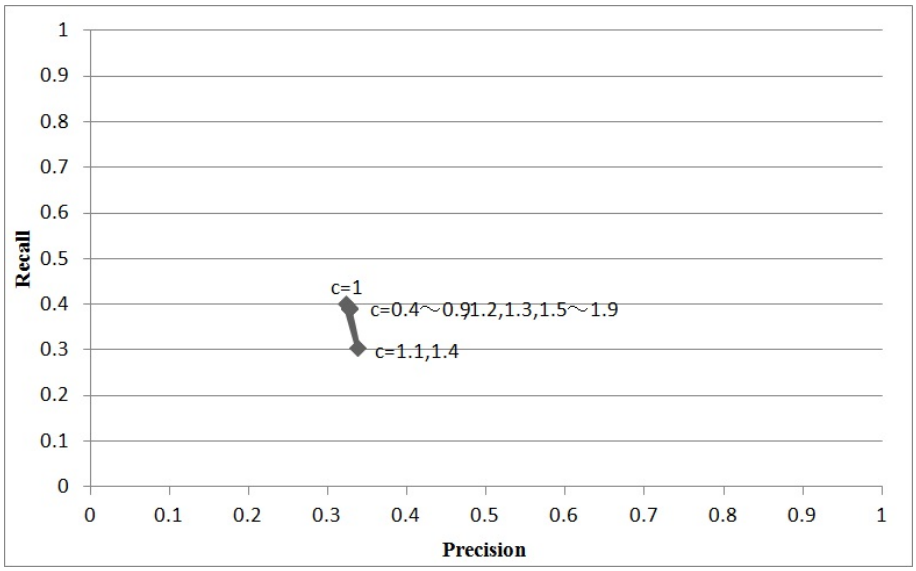

図 4.2: パラメータ c の変化と分類性能

4.3 名詞化表現による因果関係の抽出の性能に関する評価

本研究では因果関係を抽出する手掛かりとして，イベントの共参照表現としての名詞化表現を利用する．そこで，名詞化表現を手掛かりとして，潜在的に文書中に存在する因果関係の中で，どのくらいの数の因果関係が，どの程度正確に抽出できるのかを評価する．因果関係知識の評価にはよく知られたデータセットは存在しないため，人手により正解データを作成し，評価に用いる．

4.3.1 実験設定

本研究では，特定の動詞を入力として受け取り，その動詞に対する因果関係を抽出する手法を提案している．故に評価用データは，いくつかの動詞をあらかじめ選んでおき，それぞれの動詞について因果関係の正解データをラベル付けしてもらう，という形で作成する．まず，評価に用いる動詞の選択について説明する．

動詞の選択

本研究で入力として想定している動詞は，エンティティA，B を用いて “A *verb* B” と記述され，FrameNet に掲載されている必要があった．特に前者に該当するような動詞を自動的に抽出するのは難しく，コストがかかるため，手動で評価に用いる 10 個の動詞を選んだ．なお，実際には本研究では意味フレーム単位での因果関係の抽出を行うため，評価に用いられるのは 10 個の意味フレームである．この 10 個の意味フレームとそこに属する動詞，また親フレームと抽出される名詞化表現の一覧を，表 4.3 に示す．

評価用データの作成

次に，選んだ動詞それぞれに対して，因果関係の正解データを作成する．正解データの作成は，以下の手順で行う．

1. コーパスから選んだ入力動詞と，その名詞化表現を含む文書を探し，提案手法とベースライン手法 (後で説明) で 1 つ以上の因果関係が抽出できることを確認する．
2. 抽出した文書の中から，ランダムに 5 つの文書を選ぶ．
3. 選んだ文書の中で，入力動詞の出現箇所をマークしておく．

表 4.3: 評価実験に用いた 10 個の意味フレームの詳細

フレーム (親フレーム)	品詞	単語 (LU)
Attack	動詞	ambush, assail, assault, attack, bomb, bombard,
(Intentionally_affect)		charge, hit, fall, infiltrate, invade, jump, lay, raid, set,
		storm, strike
	名詞	airstrike, ambush, assailant, assault, attack, attacker,
		fire, bombardment, bombing, charge, drive-by,
		incursion, invader, infiltration, offensive, onset,
		onslaught, raid, SAF, safire, small arms fire, strike
Commerce_buy	動詞	buy, purchase
(Getting)	名詞	purchase, acquisition, procurement
Commerce_sell	動詞	auction, retail, sell, vend
(Giving)	名詞	auction, retailer, sale, charity, contribution, donation,
		gift
Finish_competition	動詞	fold, lose, show, tie, win
(Finish_game)	名詞	draw, loss, tie, victory, win
Getting	動詞	acquire, gain, get, obtain, procure, score, secure, win
(Event)	名詞	acquisition, procurement, development, event
Intentionally_create	動詞	create, develop, establish, found, generate, make,
(Creating,		produce, set up, synthesise
Intentionally_act)	名詞	creation, development, establishment, generation,
		production, synthesis, formation, issuance, act, action,
		activity, actor, agent, doing, measures, move, step
Process_stop	動詞	cease, desist, discontinue, kill, quit, shut down, stop
(Event)	名詞	cessation, discontinuation, halt, shutdown,
		development, event
Reveal_secret	動詞	admit, come forward, confess, confide, disclose, leak,
(Statement)		divulge, expose, fess up, reveal, spill beans, tip hand
	名詞	confession, divulgence, acknowledgment, admission,
		affirmation, allegation, announcement, assertion, claim,
		statement, comment, concession, conjecture, contention,
		declaration, denial, exclamation, explanation, insistence,
		mention, message, proclamation, promulgation, remark,
		pronouncement, proposal, proposition, report, avowal
Supply	動詞	equip, fix up, fuel, furnish, issue, outfit, provide,
(Giving)		provision, supply
	名詞	equipment, provision, supplier, supply, charity,
		contribution, donation, gift
Visiting	動詞	revisit, visit
(Being_located,	名詞	call, visit, twenty, whereabouts, act, action, activity,
Intentionally_act)		agent, doing, measures, move, step

表 4.4: 因果関係としてラベル付けされた動詞の数

意味フレーム	C	R
Attack	17	18
Commerce_buy	15	13
Commerce_sell	14	15
Finish_competition	12	6
Getting	20	13
Intentionally_create	6	12
Process_stop	6	21
Reveal_secret	2	12
Supply	6	5
Visiting	10	31
Total	108	146

4. 評価者に文書を渡し，文書中で入力動詞と因果関係にある動詞のラベル付けを行ってもらう．ラベル付けの際，入力動詞と，文書内のある動詞が因果関係にあるかどうかが曖昧であったり，判断が難しい場合が存在する．そこで，以下の 2 種類に分けて，因果関係のラベル付けを行ってもらう．

   - C(causality): その動詞が入力動詞と明確な因果関係にある場合．
   - R(relatedness): その動詞と入力動詞は明確な因果関係にあるとは言えないが，何らかの関係があると考えられる場合．

以上の手順で，10 個の意味フレームに対して動詞 5 個ずつ，計 50 個の文書 (ニュース記事) について，評価者によるラベル付けを行った．それぞれの意味フレーム毎に，ラベル付けされた動詞の数を表 4.4 に示す．今回実験に使う意味フレームは，因果関係が多く存在するもの (表 4.4 の上から 5 つ) と，あまり存在しないもの (表 4.4 の下 5 つ) に分かれた．

評価実験の内容

作成した正解データを用いて，以下の手順でイベント共参照表現を用いることによる因果関係の抽出の性能を評価する．

1. 入力動詞とそれを含む文書が与えられる．

2. 提案手法，またはベースライン手法を用いて，因果関係を表現する語彙パターンを作成する．

3. 各文書において，C または R でラベル付けされた動詞が語彙パターンによって捕捉できるかどうかを調べる．

4. 入力動詞と因果関係にある動詞 (C)，及び因果関係または何らかの関係にある動詞 (C+R) に対する捕捉性能の Precision，Recall を求めることで評価とする．

この実験では複数の関係とそれらが引き起こす因果関係，という本研究の最終的な出力の評価ではなく，本研究が作成する名詞化表現を主語とした語彙パターンにどの程度因果関係が記述されるか，という評価である．従って，提案手法の中で，関係の選択の部分については評価していない．この部分の評価については，次節の実験で行う．

次に，用いるベースライン手法について説明する．

ベースライン手法

提案手法に対するベースライン手法として，以下のような因果関係の抽出手法を用いる．

1. 提案手法と同じ解析を行い，入力動詞 *verb* に対してイベント “A *verb* B” と，A や B に入るエンティティのリストを抽出する．

2. Reverb システムを用いて，文書に存在する関係を全て抽出する．

3. 抽出された関係の中で，A や B をエンティティの片方，または両方に含む関係を入力動詞と因果関係にある語彙パターンとして出力する．

関連する研究では関係の間の関連性を測るために，例えば自己相互情報量 (PMI) のような尺度や，機械学習を用いる場合がほとんどであるが，今回の実験では上記のような非常に単純な手法をベースラインとして用いる．これは，純粋にイベント共参照表現を用いる場合と用いない場合の，因果関係の抽出性能やその特徴を比較・評価するためである．

4.3.2 実験結果

表 4.5 に意味フレームごとの因果関係の抽出実験の結果を示す．表中 C，R は各手法が作成した語彙パターンの中に含まれる動詞が，C，R とラベル付けされた動詞であった数である．¬(C+R) は語彙パターン中の動詞で，C と R のいずれにもラベル付けされていなかった数である．

表 4.5: 意味フレームごとの因果関係の抽出実験の結果

	ベースライン			提案手法		
意味フレーム	C	R	¬(C+R)	C	R	¬(C+R)
Attack	2	8	43	5	1	0
Commerce_buy	4	7	29	6	0	0
Commerce_sell	0	1	14	3	1	0
Finish_competition	2	3	58	5	0	0
Getting	2	1	18	6	1	2
Intentionally_create	3	1	18	1	1	4
Process_stop	0	6	23	1	1	3
Reveal_secret	0	2	37	1	0	4
Supply	1	3	49	1	0	4
Visiting	3	6	38	1	2	2
Total	17	38	327	30	7	19

表 4.6: 因果関係の抽出の性能の比較

手法	Precision	Recall	F_1
ベースライン手法 (C)	4.45%	15.74%	6.94%
ベースライン手法 (C+R)	14.40%	21.65%	17.30%
提案手法 (C)	53.57%	27.78%	36.59%
提案手法 (C+R)	66.07%	14.57%	23.87%

提案手法はベースライン手法に対して特に Precision で大きく上回る結果を得た．一方で，提案手法が非常に有効な意味フレーム (Commerce_buy や Finish_competition) と，有効でないフレーム (Reveal_secret や Supply) に分かれた．これは “C” とラベル付けされた動詞の数 (表 4.4) と一致する．この傾向はベースライン手法でも見られた．また，提案手法で抽出できた “C” とラベル付けされた動詞 30 個の中で，ベースライン手法でも抽出できたものは 8 個であった．

4.4 ルール毎の精度の評価

前節の実験で，語彙パターンの動詞の捕捉性能を調べることで，潜在的に存在する因果関係をどの程度抽出の対象にできるかを調べたが，因果関係のルール全体の評価は行っていない．そこで本実験では，作成したルール自体の正確性を調べる．また，SVM を用いることの有効性を評価する．

4.4.1 実験設定

使用する文書は前節の実験と同じものを用いる．すなわち，10 個の意味フレームに対して 5 つずつ，計 50 文書をルール抽出の対象とする．

実験の内容

提案手法を含む 4 つの手法により，50 個の文書からルールを作成する．作成したルールそれぞれに対し，2 人の評価者が以下の 4 つの基準で正誤判定を行う．

- C(causality): その動詞が入力動詞と明確な因果関係にある場合．
- R(relatedness): その動詞と入力動詞は明確な因果関係にあるとは言えないが，何らかの関係があると考えられる場合．
- ER(error in relation): 各手法が選択した関係を訂正すれば，因果関係が成り立つ場合．つまり，結果の部分を表わす語彙パターンは正しく，X と A や B との関係に誤りがある場合である．
- F(false): 語彙パターンが誤りがあり，ルールが成立しない場合．

手法

本実験では以下の 4 つの手法の比較を行う

- Reverb+Reverb
  4.3.1 節で説明した手法を利用して，例えば「“A *verb* B” → “B *verb'* X”」のような因果関係のルールを作成する．A，B の中で結果節に存在しない変数と X との関係を，関係リソースの中からその変数と，X をエンティティとして持つ “A *relation* X” のような関係を抽出[6]し，Reverb のスコアの最も高い関係を条件節に加える．最終的に,「“A *verb* B” AND “A *relation* X” → “B *verb'* X”」のようなルールを得る．

[6] ここでも提案手法と同様，X に関しては MySQL のクエリ拡張機能を利用した曖昧な検索を行う．

- Reverb+SVM
  Reverb+Reverb と同様の手法でルールを作成する．ただし，関係の選択には SVM 分類器を用いる．

- 提案手法+Reverb
  提案手法を用いて因果関係のルールを作成するが，関係の選択に Reverb のスコアを用いる．

- 提案手法+SVM
  SVM を用いた関係選択を行う，提案手法によりルールを作成する．

評価では上記 4 つの手法が作成したルールをランダムに並び変え，どの手法が作成したルールなのかは伏せた状態で評価を行う．また，提案手法で抽出できるルールの数と，Reverb を用いたベースライン手法で抽出できるルールには差があるため，公平な評価のために評価するルールの数は全ての手法で一致させる．具体的には，4 つの手法である文書からルールを抽出したとき，その中で最も少ないルール数に揃える．すなわち，その数よりも抽出したルール数が多かったものについては，ランダムにルールをその数だけ選ぶことで，数を揃える．

4.4.2 実験結果

表 4.7 に評価者 A の評価結果，表 4.8 に評価者 B の評価結果，表 4.9 に 2 人の評価の平均を示す．各表中 Accuracy(C) は抽出されたルールの中で，C(causality) と評価された割合，Accuracy(C+R) は C または R(relatedness) と判断された割合である．両者の評価結果のいずれにおいても，SVM を用いた提案手法が一番良い精度を得た．一方で，SVM を使用した際の効果は提案手法では顕著であったものの，Reverb を用いたベースライン手法ではそれほど差が無かった．

また，両者とも全ての手法において ER(関係の誤り) と判断されたルールが，語彙パターン自体の誤りと同数か，それ以上存在することが分かった．

また，Cohen のカッパ係数[7][6] を計算した結果，κ=0.0239 であった[8]．

[7] 2 人の評価者の判断の一致度を評価する．
[8] この値は，“Slight agreement”(僅かな一致) と評価される．

表 4.7: 評価者 A の評価結果

手法	C	R	ER	F	Accuracy(C)	Accuracy(C+R)
Reverb+Reverb	1	19	31	0	1.96%	39.22%
Reverb+SVM	3	19	28	1	5.88%	43.14%
提案手法+Reverb	19	14	15	8	33.93%	58.93%
提案手法+SVM	23	11	16	6	41.07%	60.71%

表 4.8: 評価者 B の評価結果

手法	C	R	ER	F	Accuracy(C)	Accuracy(C+R)
Reverb+Reverb	16	14	10	11	31.37%	58.82%
Reverb+SVM	9	18	12	12	17.65%	52.94%
提案手法+Reverb	14	20	12	10	25.00%	60.71%
提案手法+SVM	19	18	9	10	33.93%	66.07%

表 4.9: 2 人の評価の平均

手法	Accuracy(C)	Accuracy(C+R)
Reverb+Reverb	16.67%	49.02%
Reverb+SVM	11.76%	48.04%
提案手法+Reverb	29.46%	59.82%
提案手法+SVM	37.50%	63.39%

表 4.10: 抽出されたルールの数

意味フレーム	#Document	ルールの数
Attack	82214	234
Commerce_buy	7190	69
Commerce_sell	8710	30
Finish_competition	35033	268
Getting	14706	55
Intentionally_create	105806	236
Process_stop	18060	43
Reveal_secret	43281	57
Supply	24316	23
Visiting	23252	89
Total	362568	1104

4.5 コーパス全体を対象とした因果関係抽出

ここまで，2 つの実験ではコーパスから文書の数を絞り，限られた範囲での因果関係の抽出を行った．本節では，提案手法を用い，表 4.3 にある意味フレームを用いて，コーパス全体に対して因果関係抽出を行った結果を示す．

表 4.10 に意味フレーム毎に抽出できたルールの数を示す．表中 “#Document” は意味フレーム中の動詞と名詞化表現が共起した文書の数を示す．属する動詞の数，動詞と名詞化表現が抽出できる文書数によらず，意味フレーム毎に抽出できるルールにばらつきのある結果になった．

表 4.11 に実際に抽出されたルールの一例を示す．

表 4.11: 提案手法が作成したルールの具体例

意味フレーム	ルールの例
Attack	(A attack B)
	⇒ The raid killed NUMBER people
Commerce_buy	(A buy B) AND (B have office in X1)
	⇒ The acquisition would expand A 's subscriber base in X1
Commerce_sell	(A, sell, B)
	⇒ The sale would bring a profit of MONEY
Finish_competition	(A, win, B) AND (X1 be nominate for B)
	⇒ victory did not demonstrate a wave of X1
Getting	(A, acquire, B)
	⇒ The acquisition should push A 's revenues past MONEY
Intentionally_create	(A, find, B) AND (B, be a catalyst for, X1)
	⇒ The move focused attention on X1
Process_stop	(A stop B) AND (B be a country in X1)
	⇒ the action would have effect on X1
Reveal_secret	(A, leak, B)
	⇒ the report intensified the debate on A
Supply	(A, provision, B) AND (B, be a program fund by, X1)
	⇒ a provision would limit grants to X1
Visiting	(A, visit, B)
	⇒ The visit will kick off a campaign by A

第5章　議論

5.1　エラー分析と解決策

実験の結果，各動詞について知識として有用なものは多数獲得できたが，一方で多数のルールに誤りが生じた．ここではその原因を分析する．

5.1.1　イベント共参照の誤り

まず，文書中に出現した名詞化表現が，入力動詞を参照していない場合に起こる誤りがある．これは，提案手法の精度 (Precision) を下げる一因となったと考えられる．

本研究では，使用する文書をニュース記事のような比較的小さな長さのものと仮定し，同一文書内に出現する入力動詞と名詞化表現は，全て共参照関係にあると見なしていた．例えばニュース記事のタイトルに “A buys B” が記述されていた時，同一記事の中で出現する “The acquisition” は，大抵タイトルになっているイベントを参照していると考えることができる．評価実験の結果からこの仮説はある程度正しいことが推測できるが，更に精度の高い抽出を行うためには，文書中の名詞化表現が入力動詞を本当に参照しているかどうかの検証が必要である．自然言語処理の研究の中には，例えば “acquisition” という名詞表現の主語と目的語を当たることを目的とした研究もある [15]．このような研究では本研究と逆方向のアプローチにより，名詞の主語と目的語を抽出している．すなわち，本研究で抽出できる知識を利用することで，例えば “buy” と “give” は因果関係を構成することがある，ということが分かるが，このような動詞の相互関係を利用して名詞の主語と目的語を当てている．すなわち，本研究とは逆方向の研究であるため，本研究と併せてブートストラップ的なシステムを作成することができる．したがって，本研究は名詞の主語や目的語を抽出する研究にも応用可能である．

5.1.2 関係の選択の誤り

抽出したパターンは正しいが，変数 X と A や B との関係抽出において生じた誤りがある．これは提案手法の精度を下げる主原因となっており，誤りの中で 60%を占めることが実験から分かっている．

関係の選択での誤りの理由として，変数 X に文書中で元々入っていた名詞が，例えば “Nevada” のようなエンティティとなりやすい単語ではなく，単に “market” などの一般名詞が入っていたことが原因として挙げられる．例えば A や B が会社名で合った場合，もし X に “car market” のような名詞句が入っていれば “B works on X” のような関係を抽出することが想定できるが，X に単に “market” が入っていた場合，どのような関係を求めればよいのか，という問題は非常に難しくなる．

このような問題を解決するためには，SVM の特徴量の設計を改善することが最も良い解決策である．特に，“works on” のような関係を表現するパターンに関する特徴量を更に増やす必要があると考えられる．他にも，X で置き換え可能な名詞に制限を加えるなどの解決策も考えられる．

5.1.3 名詞化表現の誤り

FrameNet を利用した名詞化表現の抽出で，本研究が対象とする知識を獲得するのに相応しくない名詞化表現を抽出したことによる誤りがある．例えば，動詞 “win” において，意味フレーム “Finish_competition” には名詞 “lose” が属する．提案手法は入力動詞の拡張としてイベントの抽出に “lose” を利用し，名詞化表現として “Finish_competition” から抽出した “victory” を利用する．“lose” と “victory” は共参照関係には成り得ないのは明らかである．

これは，FrameNet 以外の外部リソースを併用する必要があることを示唆している．例えば，WordNet 等で単語同士の意味的な関連度を測ることのできるスコア関数を導入し，提案手法が動詞の共参照表現と見なした名詞へのフィルタリングを追加することにより，名詞化表現の抽出性能の向上が期待できる．

5.1.4 既存の言語処理ツールの誤り

固有表現抽出や係り受け解析，代名詞等の共参照表現の解決に失敗したことによるエラーも多数存在する．このようなエラーは精度，再現率（Recall) 低下の一因となる．

特に共参照解決は非常に難しく[1][2]，語彙パターン中の “it” や，“the company” が解決されないまま残ってしまう場合がある．また，固有表現抽出の誤りや係り受け解析が原因で，本来はパターンに必要な単語を削除されてしまったり，イベントの抽出が上手く行われないことがある．

本研究では言語処理ツールの結果はすべて正しいとは限らず，しばしば誤りを含むことをあらかじめ想定し，経験的なルールによりフィルタリングを行うことで，ツールの誤りに起因する精度の低下の防止を行っている．さらなる性能の向上のための明確な解決策は存在しないため，更なるルールの精査等で対処しなければならない．

5.2 提案手法の限界

構文解析などの結果や，イベント共参照表現の解決の結果がすべて正しかったとしても，提案手法で因果関係の抽出が誤ることがある．それは，名詞化表現が主語になる部分木が，入力動詞の定義的な説明を行っている場合に生じる誤りである．例えば，

> The acquisition was held Friday, April 7, 2006...”

このように，入力動詞が表現するイベントの付加的な情報を表現している場合は，因果関係と見なすことはできない．また，

> The acquisition reflects ...”

のような語彙パターンはイベントの結果よりは，むしろ原因を表現していると考えられる．両者とも現在の提案手法では因果関係と見なしている．将来的には語彙パターンごとの因果関係成立に対するスコア付けが必要になると考えられる．そのためにはどのような語彙パターンや構文，文脈の場合に因果関係が成立しないのか，という知識を蓄積する必要がある．どのようにデータを作り，それをどう生かすか，という点について今後の検討課題としたい．

[1] http://nlp.stanford.edu/software/dcoref.shtml

[2] 共参照解決の性能を評価する CoNLL Shared Task データセットでの F_1 値の平均が 50%台である．

5.3 語彙パターンの再利用性の向上

本研究では因果関係の結果節を変数を含む語彙パターンの形で表現することで，知識の再利用性の向上を行った．しかしながら，本研究の獲得した知識の結果節が更に原因となって別の結果を生む，という知識は現状のままでは簡単に判断することはできない．それは原因節と結果節の表現形式が異なるからである．

知識の再利用性の更なる向上を行うためには，以下のように語彙パターンを変換すればよい．

The acquisition gives A operations in X" $\Rightarrow$ get_operations_in(A, X)

このように表現すると，複数の関係が導く 1 つの関係という表現となるため，知識を連鎖的に繋ぐことも可能となる．このような変換を行うためには，例えば，

gives α β $\Rightarrow$ get(α, β)

のような変換ルールが必要となる．そもそもこのような変換ルールを作成することは，構文木の変換や動詞の意味を考慮した変換を行う必要があり，非常に難しい問題である．語彙パターン変換の例にもあるように，前置詞等が含まれる場合には特に変換ルールは難しくなる．

このような処理は単に語彙パターンの変換の問題ではなく，関係抽出における関係の構築，変換という研究にも関係すると考えられ，今後の研究課題となりうる．

5.4 名詞化表現の拡張

本研究では，動詞を受ける名詞化表現を因果関係抽出の上で非常に重要な要素として考えてきたが，そもそも本質的に辞書等からは抽出のしにくいイベント共参照関係も存在する．例えば，以下のような例である．

A buys B. The **deal** will increase....

この "deal" という名詞は単に「取引」という意味であるため，文脈から判断しなければ，買収行為を指していることは分からない．このような表現は FrameNet のようなリソースを使用しても抽出することは困難である．そこで，このような表現を収集し，提案手法の再現率の向上を行うようなものとして，図 5.1 のようなシステムを考える．

このシステムでは一度抽出した語彙パターンの中で，名詞化表現が記述されている部分を変数に置き換え，名詞化表現の検索用パターンとして用いることで，ブートストラップ的なアルゴリズムにより更なる名詞化表現の収集を行うことができる．もちろん，この

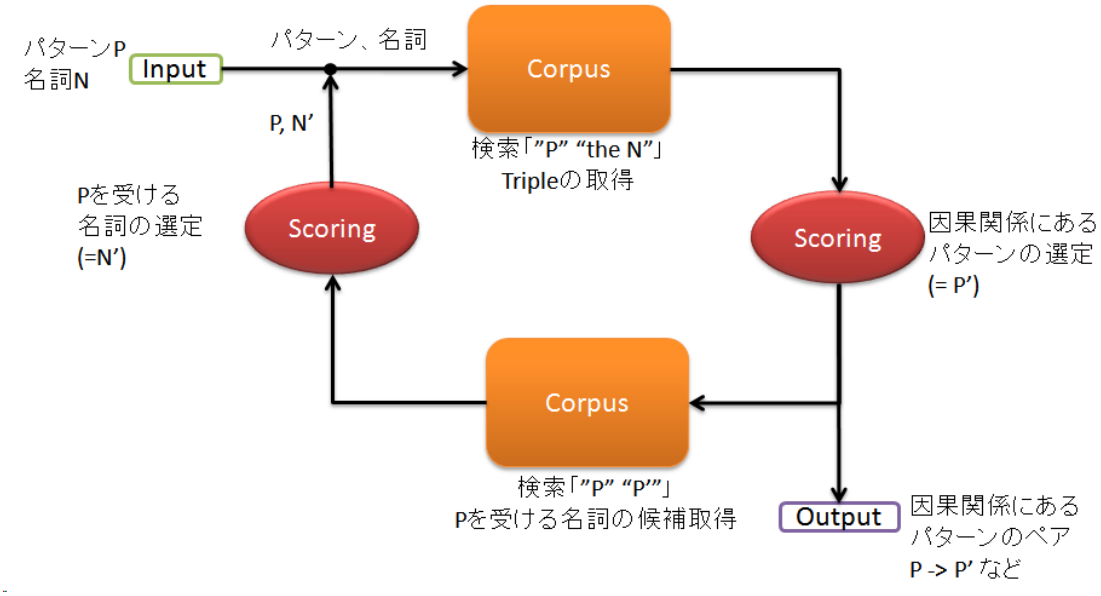

図 5.1: 再帰的に名詞化表現を獲得するシステム

ようなシステムを構築するためには，ある名詞が動詞の共参照表現となりうるのか，ということを計算できるような指標が必要であり，これは今後の研究課題である．

5.5 入力動詞の制約

本研究では入力動詞を，イベントの抽出が適切に行うことのできるもの，すなわち “A *verb* B” で表現できるもの，という制約を設けていた．因果関係の抽出の観点から言うと，例えば，

Michael Jackson died. The death causes ...

のように，“A *verb*” というエンティティが片方のみのイベントが原因となる因果関係も考えることができる．また，入力も動詞とは限らず，

Japan took part in World War I. The participation causes...

という，熟語的な表現もイベント抽出の対象とすることができる．

本研究では，このようにいくつか存在するイベント表現の中の 1 つを，イベント抽出の対象とした．上記の 2 例のうち，前者は比較的容易に抽出できると考えられるが，イベントの情報量がエンティティ 1 つ分欠落することになるため，精度の面で検証が必要である．

後者については，熟語表現に対して，それを参照する名詞化表現の抽出が課題となる．これは (知る限り) 既存のリソースでは獲得できないため，前節で述べたようなシステムを利用して動詞と名詞化の拡張を行う必要がある．ただし，動詞の拡張まで含めると 3 重のブートストラップとなるため，それによる精度の低下を招かないような手法を検討する必要がある．

5.6 既存研究との統合

本研究では特にイベントの共参照表現が記述する因果関係に注目し，抽出の対象とした．実験の結果からも分かるように，因果関係が記述されるのはイベント共参照表現の周りだけではない．本研究では精度で既存研究よりも優れている一方で，再現率では既存研究に劣る．また一方で，既存研究は因果関係の認識に留まるものも多く，変数を導入した再利用性の高い知識へと変換することが行われていない．本研究のような語彙パターンへの抽象化を既存研究に導入し，本研究で獲得した知識と統合することで，本質的に存在する因果関係の知識をより高精度・高再現率で抽出できることが期待できる．

第6章　おわりに

6.1　結論

本稿では，ある特定の動詞が表現するイベントと因果関係にある知識を獲得するため，イベント共参照表現としての名詞化表現を用いる手法を提案した．特に本研究では単に因果関係の識別を行うだけでなく，変数の入った再利用可能な語彙パターンの形で知識を表現することで，より高度な因果関係知識の獲得を行った．

提案手法は人手により作られた評価用データを用いて評価し，従来の関係抽出をベースとした手法よりも高精度に因果関係を獲得でき，かつ従来の研究では獲得することのできなかった因果関係の知識を多く獲得できることを示した．

6.2　今後の展望

提案手法は今後，イベントの抽出範囲や名詞化表現の拡張により，更なる性能の向上が期待できる．また，既存の研究との統合により，文書中に潜在的に存在する因果関係の抽出を行う研究に対して，大きな貢献を行うことも期待できる．

将来的に，本研究が大量に存在する情報の構造化と，応用的な検索システムの開発に貢献し，更に「賢い」システムの構築に役立つことを望む．

参考文献

[1] Azad Abad, Luisa Bentivogli, Ido Dagan, Danilo Giampiccolo, Shachar Mirkin, Emanuele Pianta, and Asher Stern. A resource for investigating the impact of anaphora and coreference on inference. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2010.

[2] Brandon Beamer and Roxana Girju. Using a bigram event model to predict causal potential. In *Computational Linguistics and Intelligent Text Processing, 10th International Conference*, pp. 430–441, 2009.

[3] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of focused entailment graphs. In *Procedings. of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1220–1229, 2010.

[4] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 610–619, 2011.

[5] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases, International Workshop WebDB'98*, pp. 172–183, 1998.

[6] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, Vol. 70, No. 4, pp. 213–220, 1968.

[7] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classi cation, and Recognising Tectual Entailment*, Vol. 3944, pp. 177–190, 2006.

[8] Quang Xuan Do, Yee Seng Chan, and Dan Roth. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 294–303, 2011.

[9] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 3–10, 2011.

[10] Luis Gravano Eugene Agichtein. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pp. 85–94, 2000.

[11] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545, 2011.

[12] Charles J. Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Vol. 280, No. 1, pp. 20–32, 1976.

[13] Charles J. Fillmore and B. T. S. Atkins. Framenet and lexicographic relevance. In *Proceedings of the First International Conference on Language Resources and Evaluation*, 1998.

[14] Christiane Fellbaum Derek Gross George A. Miller, Richard Beckwith and Katherine Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, Vol. 3, pp. 235–244, 1990.

[15] Matthew Gerber and Joyce Yue Chai. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1583–1592, 2010.

[16] Roxana Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pp. 76–83, 2003.

[17] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pp. 539–545, 1992.

[18] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 200–209, 1999.

[19] Xiaojiang Liu Bo Zhang Jun Zhu, Zaiqing Nie and Ji-Rong Wen. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, pp. 101–110, 2009.

[20] Dekang Lin and Patrick Pantel. Dirt - discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2001.

[21] Peter LoBue and Alexander Yates. Types of common-sense knowledge needed for recognizing textual entailment. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 329–334, 2011.

[22] Jeffrey Bigham Andrei Lifchits Marius Pasca, Dekang Lin and Alpa Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *Proceedings of the 21st national conference on Artificial intelligence*, pp. 2670–2676, 2007.

[23] Stephen Soderland Michael J. Cafarella, Doug Downey and Oren Etzioni. Knowitnow: fast, scalable information extraction from the web. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 563–570, 2005.

[24] Stephen Soderland Matt Broadhead Oren Etzioni Michele Banko, Michael J. Cafarella. Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pp. 2670–2676, 2007.

[25] Marco Pennacchiotti Patrick Pantel. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 113–120, 2006.

[26] Mehwish Riaz and Roxana Girju. Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In *Proceedings of the 4th IEEE International Conference on Semantic Computin*, pp. 361–368, 2010.

[27] Stijn De Saeger, Kentaro Torisawa, Masaaki Tsuchida, Jun'ichi Kazama, Chikara Hashimoto, Ichiro Yamada, Jong-Hoon Oh, Istvan Varga, and Yulan Yan. Relation acquisition using word classes and partial patterns. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 825–835, 2011.

[28] Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *Proc. of the 2010 Conference on EMNLP*, pp. 1088–1098, 2010.

[29] Yizhou Sun, Kunqing Xie, Ning Liu, Shuicheng Yan, Benyu Zhang, and Zheng Chen. Causal relation of queries from temporal logs. In *Proceedings of the 16th International Conference on World Wide Web*, pp. 1141–1142, 2007.

[30] Patrick Suppes. *A probabilistic theory of causality.* North-Holland Pub. Co. (Amsterdam), 1970.

[31] 大友謙一, 柴田知秀, 黒橋禎夫. 述語項構造の共起情報と節間関係の分布を用いた事態間関係知識の獲得. 言語処理学会第 17 回年次大会 (NLP2011), 2011.

発表文献

自著論文

1. Shohei Tanaka, Naoaki Okazaki and Mitsuru Ishizuka. Learning Web Query Patterns for Imitating Wikipedia Articles. In *Proc. of 23rd Int'l Conf. on Computational Linguistics (COLING 2010) – Poster Volume*, pp. 1229-1237, 2010

2. 田中翔平, 岡崎直観, 石塚 満. Wikipedia を教師データに用いた要約文書収集クエリパターンの学習. 人工知能学会論文誌, Vol.26, No.2, pp.366-375, 2011

3. 田中翔平, 岡崎直観, 石塚 満. カテゴリ情報を考慮した Wikipedia からの含意関係の抽出. 第 25 回人工知能学会全国大会, 2011

4. 田中翔平, 岡崎直観, 石塚 満. イベント共参照関係を利用した因果関係知識の獲得. 第 74 回情報処理学会全国大会 (発表予定), 2012

共著論文

1. Ken-ichi Yokote, Shohei Tanaka and Mitsuru Ishizuka. Effects of Using Simple Semantic Similarity on Textual Entailment Recognition. In *RTE-7(Recognizing Textual Entailment) Workshop Note at Text Analysis Conference(TAC2011)*, 2011

2. 横手健一，田中翔平，ダヌシカ ボレガラ，石塚満. テキスト含意認識に有用な概念意味情報. 第 74 回情報処理学会全国大会 (発表予定), 2012