Wikipedia と WordNet を用いたテラバイト級 Web コーパスからのクラス間関係抽出

白川　真澄† 　中山浩太郎†† 　荒牧　英治†† 　原　隆浩† 　西尾章治郎†

† 大阪大学大学院情報科学研究科
〒 565-0871 大阪府吹田市山田丘 1-5
†† 東京大学知の構造化センター
〒 113-8656 東京都文京区本郷 7-3-1
E-mail: †{shirakawa.masumi,hara,nishio}@ist.osaka-u.ac.jp, ††nakayama@cks.u-tokyo.ac.jp,
†††eiji.aramaki@gmail.com

あらまし　本研究では，Wikipedia の記事と WordNet のクラスを用いて，大規模な Web コーパスからドメイン非依存でクラス間の関係を抽出する手法を提案する．高い精度を維持しつつ多種多様なドメインのクラス間の関係を網羅的に抽出するためには，関係の絶対数を増やすことが不可欠であり，そのためには膨大なテキストを解析する必要がある．そこで提案手法では，処理速度を優先するため，自然言語処理技術を一切用いずにテキストコーパスから語句を抽出し，2 つの語句間に出現する文字列を関係パターンとして取得する．そして，抽出した語句間の関係から，Wikipedia の記事を介して WordNet のクラスに置き換えることで，クラス間の関係を取得する．また，出現確率を考慮したスコアリングによりクラス間の関係のスコア（確信度）を定義することで，適当な粒度でクラス間の関係を取得する手法を提案する．Hadoop 環境を利用して 10 億以上の Web ページ（約 15TB）から 1 億以上のクラス間の関係を抽出し，様々な観点から評価を行い，提案手法の有効性，および抽出したクラス間の関係の有用性を確認した．

キーワード　関係抽出，Wikipedia，WordNet，YAGO

1. はじめに

人がテキストの意味を理解する背景には**概念（concept）**あるいは**クラス（class）**が存在する．概念あるいはクラスとは，事物を抽象化し，共通する意味や性質を表したものであり，事物そのものを表すエンティティ（entity）あるいはインスタンス（instance）とは区別される．たとえば,「大阪大学」や「東京大学」がある事物を表すエンティティであるのに対し,「大学」や「教育機関」はこれらのエンティティの共通の側面を表した概念である．また,「大阪大学」と「大阪城」というエンティティに対しては「大阪に存在する建造物」という概念が存在しうる．このように，1 つのエンティティには様々な上位の概念が存在し，他のエンティティとの共通点や相違点およびその他の関係は多くの場合，この概念を介して表現される．

心理学者 Gregory Murphy が自身の著書 [14] で「概念は我々の心的世界を結び付ける接着剤である（concepts are the glue that holds our mental world together）」と述べていることからも分かるように，概念は言葉の意味を理解するためのカギとなるものである．また，彼は「それら（概念）によって我々は新しい物や事象を認識・理解できるようになる（they enable us to recognize and understand new objects and events）」と述べている．実際，我々人間はテキスト中に，あるエンティティを意味する語句を観測したとき，それを概念に置き換えることでテキストの意味を把握しようとする．これにより，テキスト中に自分の知らない語句が含まれていても，テキストの意味を理解できることがある．たとえば,「シャラポワがエラニを破る」という文を見たとき，シャラポワがテニス選手であることを知っていれば，エラニが何かを知らなくてもそれがテニス選手であることを推測でき，結果としてこの文の意味を把握できる．これは，我々が日々の経験から，エンティティを概念（クラス）に置き換えながら関係を学習しており，テニス選手が破るのは同じくテニス選手であるといったクラス間の関係があることを学んでいるためである．

このようなクラスを利用したテキストの内容の把握をコンピュータに行わせることを目指し，筆者らは先行研究 [21] において，テキストコーパスから，語句間やエンティティ間ではなく，クラス間の関係を抽出してきた．先行研究では，人がクラス間の関係を学習する方法にならい，テキストから関係を抽出する際に語句をクラスに置き換えてから関係を抽出する手法を提案している．また，実際に 30GB 程度の Wikipedia の全記事をテキストコーパスとして関係抽出を行い，数百万規模でクラス間の関係を抽出している．しかし，Wikipedia 内のテキストに限定して関係を抽出しているため，関係の種類に偏りが多いという問題があった．Web 上に出現しうる関係を網羅するためには，Web の様々なドメインの膨大なテキストデータを対象として関係を抽出する必要がある．

そこで本研究では，先行研究で提案した手法を拡張し，10 億以上の Web ページというテラバイトスケールのテキストコー

パスから現実的な処理時間でクラス間の関係を抽出する手法を提案する．提案手法では，膨大な量のテキストを高速に処理するため，テキストから語句間の関係を抽出する際，自然言語処理技術を用いず，文字列処理のみを適用する．具体的には，トライ辞書を用いて語句を検出し，語句間に出現する文字列を関係パターンとして抽出する．これにより，膨大な Web コーパスからの関係抽出を比較的高速に処理できる．また，自然言語処理技術を用いないことで，英語以外の様々な言語にも実装コストをかけずに応用できる可能性がある．一方で，抽出した関係の精度が低下するという問題が発生する．この問題に対しては，関係の出現頻度をもとに確率的なスコアを与える手法を提案することにより，できる限り精度を維持する．本研究では，提案手法の有効性を確認するために，Hadoop 環境を用いて約 15TB という規模の Web コーパスからクラス間の関係抽出を行い，1 億以上のクラス間の関係を抽出している．

2. 関 連 研 究

テキストから語句間の関係を抽出する研究はこれまで数多く行われてきたが，Web の普及に伴い，大規模な Web コーパスを対象としたドメイン非依存の関係抽出手法が注目を集めてきた．Etzioni らの KnowItAll [6] はドメイン非依存で関係抽出を行った代表的な例である．KnowItAll では，関係抽出のためのパターン（例 : capitalOf 関係に対して「is the capital of」など）を用いて関係抽出を開始し，得られた出力を新たな入力として反復を行うブートストラッピング法 [17] によって新たなパターンを学習することで，関係タプル（例 : Tokyo, capitalOf, Japan の 3 つ組）の数を拡大していく．KnowItAll では Web 検索クエリを用いて Web ページを取得するため処理に時間がかかるが，TextRunner [1] は KnowItAll の非効率さを改善するため，Web コーパスをあらかじめ全て取得してから処理を行う．また，彼らは文献 [7] において，抽出した関係タプルが実際の Web の文書中にどのような構文を伴って出現するかを検証することにより，精度および網羅性の向上を図っている．WOE [20] では，Wikipedia を信頼できる情報として用いることで，TextRunner よりも高い精度で関係抽出を達成している．Bollegala らの Relational Duality [3] では，関係を表す方法として，関係パターン（例 :「is the capital of」）と語句ペア（例 :「Tokyo」「Japan」）の 2 つの側面があることを利用し，完全に教師なしでの関係抽出を実現している．これらの Web コーパスからの関係抽出手法では，1) いかに多くの事実関係を 2) 精度良く抽出するか，という 2 点に注視しているが，アプリケーションにおいて未知の語句や関係に対応するためには，クラス間の関係を充実させる必要がある．本研究では，未知の語句や関係に対する推測能力をコンピュータに持たせることを目標とし，エンティティ間の関係ではなくクラス間の関係の抽出を第一の目的としている．

また最近では，本研究と同様にクラスに注目した大規模かつドメイン非依存な関係抽出も行われている．NELL [5], [13] では，名詞句がどのようなクラスを持っているか，およびどのようなクラス間の関係があるかを同時に抽出している．ReVerb [8] は動詞句に限定して語句間の事実関係を抽出しており，現在公開しているシステムでは 5 億の Web ページから抽出した関係に，Freebase [2] のタイプ（クラス）を付与することで，どのようなクラス間で関係があるかを表現している．PATTY [15] では，YAGO [18] を用いて Wikipedia の WordNet [9] のクラス間の関係を抽出し，関係パターンのクラスタリングを行っている．本研究では WordNet のクラス間の関係を抽出することを目的としており，特に PATTY との類似性が高いが，PATTY とは異なり，完全にクラス間の関係抽出のみを対象としたアプローチをとっている．また，本研究の提案手法は，構文解析や形態素解析などの自然言語処理技術を一切用いない手法であるため，上記の研究で提案された手法よりも高速に機能し，また，英語以外の言語にも実装コストをかけずに応用できる可能性が高い．

3. クラス間の関係抽出手法

本研究では，テキストから関係を抽出する際に，語句をクラスに置き換えることで，クラス間の関係を抽出する手法を提案する．以下ではまず，人がどのように概念（クラス）を利用して関係を学習し，推測するかについて述べた後，それを模倣したクラス間の関係抽出手法について詳述する．なお，本研究における提案手法は先行研究 [21] の手法を拡張したものであるが，大量のテキストを処理するために文字列処理のみを用いて速度を重視している点，多言語展開を可能とするため Freebase [2] ではなく WordNet [9] を用いている点で大きく異なる．また，次章では，得られた関係のスコア（確信度）を確率的に決定し，より明確な意味を持つ関係に高いスコアを与える手法を提案する．

3.1 人が行う関係の学習と推測

概念（クラス）とは，あるエンティティのまとまりを抽象化し，共通する意味・性質を表したものである．たとえば,「東京大学」や「京都大学」,「大阪大学」といったエンティティ群に対し，共通した意味として「大学」や「教育機関」といったものが概念となる．また,「大阪大学」「大阪城」「通天閣」といったエンティティ群に対しては，概念は「大阪に存在する建造物」などが該当する(注1)．1 つのエンティティには様々な上位の概念が存在し，他のエンティティとの共通点や相違点およびその他の関係は多くの場合，この概念を介して表現される．概念を介して関係を学習することで，人は未知の事物を認識・理解することができる [14].

以下では，人がどのように概念間の関係を学習するかについて簡単に説明する．たとえば,「イチローがヤンキースに移籍する」という文があったとする（図 1）．人はこの文を観測したとき,「イチロー」,「ヤンキース」といったエンティティを概念「野球選手」,「球団」などを通して認識する．このとき，単に「イチローがヤンキースに移籍する」というエンティティ間の関係のみならず,「野球選手が球団に移籍する」という概念間の

(注1)：なお，組織としての大阪大学と単なる建造物としての大阪大学は別のエンティティであるとする見方もあるが，ここでは区別しないものとする．

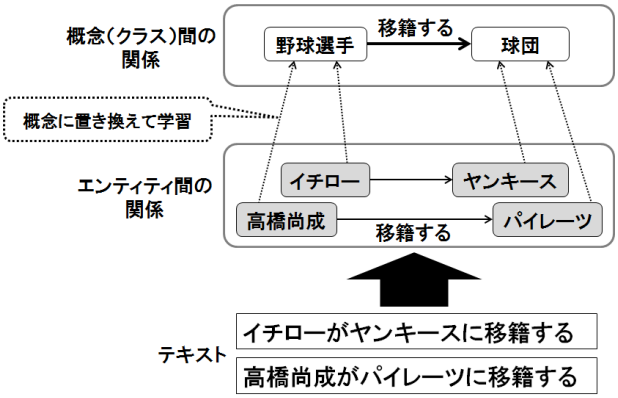

図 1　概念（クラス）間の関係の学習

関係が存在しうることも学習する．また，別の文「高橋尚成がパイレーツに移籍する」を観測したときにも，同様の関係を学習する．同じ概念間の関係をより多く観測すればするほど，その関係が一般的であると認識するようになる．一度概念間の関係を学習すれば，以降は未知の語句に対しても，学習した関係をもとにその語句の意味を推測できるようになる．たとえば，「モラレスがマリナーズに移籍する」という文に対し，モラレスが何かを知らない人でも，モラレスが野球選手であるという推測が可能である．

このように，文章を観測するたびにエンティティを概念に置き換えることで，人は概念間の関係を学習していく．もちろん，実際には人はこれよりもはるかに複雑なプロセスを経て言葉の意味を理解・学習していると考えられるが，根本的には，こうした概念への置き換えが関係の学習において重要な役割を果たしている．

コンピュータがクラス（概念）間の関係を学習する際にこのような概念への置き換えを用いることの利点としては，実際のテキストに出現するような関係を取得できることが挙げられる．クラス間の関係を抽出するための手法として，直接クラスを意味する語句が出現するテキストから関係を抽出する方法が考えられる．しかし，実際のテキストでは，直接クラスを意味する語句が出現する関係は，クラスに属するエンティティを意味する語句が出現する関係と比較すると少ない．たとえば，2013 年 2 月 15 日時点では Web のテキストには「tennis player beat tennis player」が 2 件，「person graduated from educational institution」にいたっては全く存在していなかった(注2)．一方，「Maria Sharapova beat Sara Errani」や「George W. Bush graduated from Yale University」など，具体的なエンティティを伴う関係は多くのテキストに出現していた．このことから，クラスに属するエンティティを意味する語句が出現する関係を抽出し，語句をクラスに置き換えることで，実際のテキストに出現する語句（あるいはエンティティ）間の関係を網羅できるようなクラス間の関係を抽出できると考えられる．

3.2　手法の概要

前節で述べた，人が行っている概念（クラス）間の関係の学習方法にならい，本研究では，テキストから関係を抽出する際に，語句をクラスに置き換えて関係を抽出する手法を提案する．提案手法では，入力として Web コーパスを与えると，クラス間の関係とその出現頻度が出力として得られる．

提案手法の処理の流れを図 2 に示す．まず，Web コーパスに出現する語句をトライ辞書によって抽出し，語句間に出現する文字列パターンを関係として取得する．その後，語句をクラスに置き換えることで，クラス間の関係を抽出する．以下では提案手法で使用する前提知識，および図 2 の各処理について説明する．

3.3　使用する前提知識

本研究では，自然言語処理技術を用いない代わりに，既に構築されている知識を利用してテキストから語句を抽出しクラスに置き換えることで，様々なドメインのクラス間の関係を抽出する．具体的には，提案手法では Wikipedia(注3)，WordNet [9]，および YAGO [11], [18] の知識を利用する．

Wikipedia は現時点において，世界最大規模のコンテンツ量を誇る Web 事典である．Wikipedia では，幅広い分野について，一般的なエンティティから新しいエンティティに至るまで記事が網羅されており，その記事（エンティティ）数は，2013 年 1 月時点において，最も多い英語版で 411 万，日本語版で 83 万である．また，本研究で Wikipedia を用いる別の理由として，Wikipedia のアンカーテキストが語義曖昧性解消（語句からエンティティへの置き換え）のリソースとして活用できることが挙げられる．さらに，Wikipedia が世界中の様々な言語で展開されていることも理由の一つである．同じエンティティに関する記事どうしが言語間リンクでつながってることから，多言語展開が実現可能となる．

WordNet は心理学に基づく包括的な概念体系を有しており，WordNet で定義されているクラス（WordNet ではシンセットと呼ばれる）を基準とした関係を定義することは，クラス間の関係抽出において重要な第一歩となりうる．提案手法では，語句間の関係において，Wikipedia のエンティティを介して語句を WordNet のクラスに置き換える．ここで，語句から Wikipedia のエンティティへの変換は Wikipedia の情報のみを用いて実行できるが，Wikipedia のエンティティから WordNet のクラスへの変換は，Wikipedia と WordNet をそのまま用いただけでは実行できない．そこで本研究では，YAGO で定義されている上位下位関係の情報を利用する．YAGO の上位下位関係の精度は 98%を超えているため，これを利用することで Wikipedia のエンティティから WordNet のクラスへの精度の高い置き換えが可能となる．

3.4　トライ辞書による語句間の関係抽出

提案手法では，Web コーパスが入力として与えられたときに，はじめの処理として語句（Wikipedia のアンカーテキスト）を抽出する．そして，2 つの語句が近接して出現する箇所から，語句の間に出現する文字列パターンを抽出し，語句間の関係を抽出する．大量のテキストを入力とした関係抽出では，精度向

(注2)：Web 上のテキストの検索には Google（http://www.google.com/）を使用した．

(注3)：http://www.wikipedia.org/

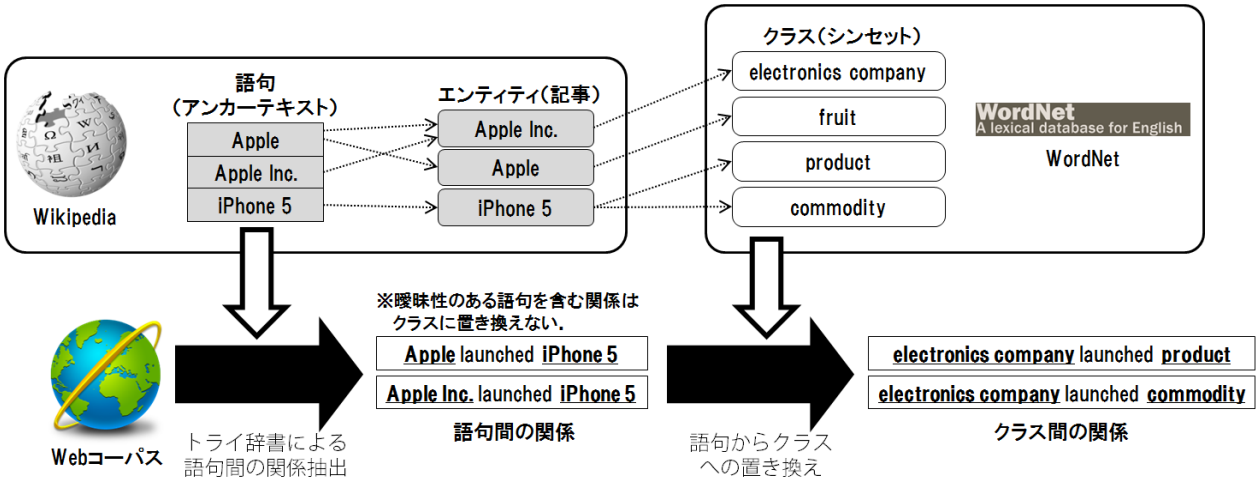

図 2　提案手法の処理の流れ

上のため，一般的に構文解析あるいは形態素解析を用いてヘッドワード（名詞句の意味の中心となる部分）を検出し，語句間の関係を抽出する．先行研究 [21] においても形態素解析を用いた手法により語句間の関係を抽出しているが，本研究では，そのような自然言語処理技術を用いず，文字列処理のみを行う．これは，本研究では様々なドメインの多様な関係を網羅するために，膨大なテキストを対象とし，抽出できる関係の絶対数を増やす必要があるためである．現状では，Web 上に存在するテキストの量は，自然言語処理技術を適用できる量を圧倒的に凌駕しているため，本研究ではあえて自然言語処理処理技術を使用せずに処理速度を重視したアプローチを採用する．なお，次章では，得られた関係のスコアリングにより，精度をできる限り維持することを目指す．

本研究では英語を対象としているため，具体的にはまず，Web ページに対して英単語 the を用いた簡単な言語判定を行い，Web ページが英語で記述されているか否かを判定する．そして，Wikipedia のアンカーテキストを語句としてトライ辞書を構築し，これを用いて各 Web ページから語句が出現する箇所を検出する．その後，2 つの語句が近接して出現する箇所から，語句間に出現する単語（最大 K 語）を，関係を表すパターンとして抽出し，語句間の関係を得る．なお，対象言語を英語としているため，ここでは英単語と特定の記号（「'」「,」「-」）のみから成る関係のみを抽出する．本研究では 12Dicts(注4) と呼ばれる英単語リストを用いている．提案手法は自然言語処理技術を用いず，また，多言語で展開されている Wikipedia を利用しているため，同様のアプローチにより，英語以外の多くの言語でも語句間の関係抽出が可能である．

Wikipedia のアンカーテキストのみを抽出対象の語句としているのは，次節で後述するように語句を Wikipedia のエンティティ（記事）および WordNet のクラスに置き換えるためである．Wikipedia の記事の編集方針の一つに「ウィキ化 (wikification)(注5)」というものがあり，記事中に登場する語句とそれが意味する記事（エンティティ）をハイパーリンクにより繋げることで，Wikipedia を整理する役目を持っている．そのため，これらの語句（アンカーテキスト）は全て Wikipedia のエンティティに置き換えることが可能となる．

(注4)：http://wordlist.sourceforge.net/
(注5)：http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Wikify

3.5　語句からクラスへの置き換え

提案手法では，抽出した語句間の関係において，語句をクラスに置き換えることでクラス間の関係を得る．ここで問題となるのが，曖昧性のある語句の処理である．すなわち，語句をクラスに置き換える際に，語義曖昧性解消（語句からエンティティへの置き換え）を行う必要がある．しかし，語義曖昧性解消は（手法にもよるが）トライ辞書による語句抽出や形態素解析などと比較して重い処理であるため，今回のような膨大なテキストを処理する場合，適用が難しい．

そこで提案手法では，曖昧性のない語句のみを対象とすることで語義曖昧性解消の問題を回避する．これは，膨大なテキストからの関係抽出では，時間をかけて精度を重視した処理を行うよりも，簡単な処理だけで高い精度を達成できそうな箇所のみから関係を抽出するアプローチをとるほうが効率的であるためである．なお，将来的には，曖昧性のある語句についても処理可能な手法を検討する．

曖昧性のない語句を判別するため，Milne らの研究 [12] などで用いられている Wikipedia のアンカーテキストとエンティティの対応関係を利用する．前述の「ウィキ化」のとおり，Wikipedia では記事中に登場する語句（アンカーテキスト）とそれが意味するエンティティ（記事）がハイパーリンクにより繋がっている．そこで，ある語句を見たとき，それがリンクされている記事に対し，出現頻度に応じた確率を割り当てる．具体的には，アンカーテキストを t，t のリンク先の記事を e，t のリンク先の記事の集合を E_t とし，以下の式によりリンク確率 $P(e|t)$ を算出する．

$$P(e|t) = \frac{CountLinks(t,e)}{\sum_{e_i \in E_t} CountLinks(t,e_i)} \quad (1)$$

$CountLinks(t,e)$ はアンカーテキスト t から記事 e へのリンク数である．提案手法では，$P(e|t)$ が 95%以上の確率である場合，t には曖昧性がないと判断し，t を e に置き換える．それ以外の場合は t には曖昧性があるとみなし，t を含む関係におい

ては，語句からクラスへの置き換えを行わない．

また，Wikipedia のエンティティから WordNet のクラスへの置き換えでは，YAGO の上位下位関係の知識をそのまま利用する．これらの処理により，語句間の関係から，最終的にクラス間の関係，およびそれらの出現頻度を取得できる．

4. クラス間の関係のスコアリング手法

提案手法により，クラス間の関係とその出現頻度が出力として得られるが，これをそのままスコアとして用いると，一般的なクラス（「person」や「group」など）がスコアの上位に出現しやすくなる．これは，単純に一般的なクラスを持つエンティティの数が相対的に多いことに起因する．しかし，本研究では，関係の種類に適した粒度のクラスを用いることで，より人が学習するクラス間の関係に近い状態で関係を定義することを目指している．たとえば，関係の種類に適した粒度のクラスの例として，「person wrote X」という関係においては，X に対して書き物全般を表す「writing」といったクラスを定義する一方，「lawyer wrote X」という関係においては，法律家が書く物として X に「law」などのより詳細な書き物のクラスを定義するほうが人にとっては自然な認識であると考えられる．また，「musician wrote X」という関係では，X に対して「song」といったクラスが当てはまると考えられる．このとき，単純に出現頻度をスコアとして用いた場合，人の間で伝達されるあらゆるものという意味として「communication」といった一般的なクラスがスコアの上位に出現しやすくなる．これは関係としては誤りではないが，「communication」よりも限定的な意味のクラスを用いても関係を定義できるため，クラスの粒度としてはあまりふさわしくないと考えられる．関係の種類に適した粒度のクラスを取得するためには，その関係にのみ出現していることが重要な指標となる．

そこで本研究では，クラスの粒度を考慮してクラス間の関係を定義するため，関係（関係パターンとその両側のクラス）の出現頻度だけでなく，クラスあるいは関係パターンの単体での出現頻度を考慮したスコアリング手法を提案する．具体的には，自己相互情報量（PMI），ローカル PMI（LPMI）[10]，正規化 PMI（NPMI）[4] をそれぞれ導入する．これらの指標を用いることで，ある関係においてどれだけ偏って（関連して）出現しているかを算出できる．

PMI は，クラスあるいは関係パターンがそれぞれ単体で出現する確率と，それらが同時に出現する確率を用いたスコアリング手法である．なお，ここでいう出現確率とは，あるクラスあるいは関係パターンの出現頻度を，全体の出現頻度の和で除算した値である．クラスと関係パターンが独立に出現している場合，クラスと関係パターンの同時出現確率は，クラスと関係パターンそれぞれの単体での出現確率の積で表される．一方，クラスと関係パターンが関連して出現している場合，同時出現確率は単体での出現確率の積に対して大きくなる．PMI では，より強く関連して出現するクラスと関係パターンに対して高いスコアを与える．

まず，関係パターンとその片側のクラスの二者間について考える．クラスを c，関係パターンを r としたとき，PMI は以下の式により算出される．

$$PMI_{(2)}(c,r) = log_2\Big(\frac{P(c,r)}{P(c)P(r)}\Big) \tag{2}$$

$P(c,r)$ はクラスと関係パターンの同時出現確率，$P(c)$ および $P(r)$ はそれぞれクラスの出現確率，関係パターンの出現確率である．また，関係パターンと両側のクラスの三者間に対しても，同様の考え方により PMI を算出する．関係パターンの両側のクラスをそれぞれ c_L，c_R，関係パターンを r とすると，これら三者間の PMI は以下の式により表される．

$$PMI_{(3)}(c_L,r,c_R) = log_2\Big(\frac{P(c_L,r,c_R)}{P(c_L)P(r)P(c_R)}\Big) \tag{3}$$

PMI では，単体での出現頻度が低い事象に対して極端なスコアを与えてしまうことが知られている [4]．この問題を解決するため，LPMI や NPMI といったスコアリング手法が提案されている．LPMI では，二者間あるいは三者間の関連の強さを表す PMI に対し，それらが同時に出現する頻度を掛け合わせることでスコアを調整する．二者間については，クラス c と関係パターン r が同時に出現する頻度を $Freq(c,r)$ とすると，

$$LPMI_{(2)}(c,r) = PMI_{(2)}(c,r) \cdot Freq(c,r) \tag{4}$$

により LPMI が算出される．三者間の場合，関係パターンの両側のクラス c_L，c_R と関係パターン r が同時に出現する頻度を $Freq(c_L,r,c_R)$ とすると，LPMI は以下の式により計算できる．

$$LPMI_{(3)}(c_L,r,c_R) = PMI_{(3)}(c_L,r,c_R) \cdot Freq(c_L,r,c_R) \tag{5}$$

NPMI は，PMI の最大値が 1 となるよう正規化した値であり，単体での出現頻度によるスコアの偏りを軽減できる．二者間あるいは三者間の NPMI はそれぞれ以下の式により表される．

$$NPMI_{(2)}(c,r) = \frac{PMI_{(2)}(c,r)}{-log_2P(c,r)} \tag{6}$$

$$NPMI_{(3)}(c_L,r,c_R) = \frac{PMI_{(3)}(c_L,r,c_R)}{-2log_2P(c_L,r,c_R)} \tag{7}$$

5. 10 億ページからの関係抽出

先行研究 [21] では，30GB 程度の Wikipedia の全記事に対してクラス間の関係抽出を行ったが，本研究では 10 億以上の Web ページ（約 15TB）を対象としてクラス間の関係抽出を行う．3 章で説明した手法を用いてクラス間の関係を抽出した後，4 章で説明したスコアリングを行い，様々な観点から評価を行う．

解析対象とする Web コーパスとして，Common Crawl コーパス(注6)を利用した．Common Crawl コーパスは，Amazon Web Services（AWS）で使用できる Web コーパスであり，全

(注6)：http://commoncrawl.org/

表 1　小規模なテキストデータに対する処理時間の比較

形態素解析を用いた手法	8,953 秒
トライ辞書のみを用いた手法	1,921 秒

表 2　Web コーパスから抽出したクラス間の関係数

本研究	**170,701,978**
先行研究 [21]	7,409,974
PATTY [15]	350,569

サイズは 81TB である．本研究ではそのうち，15TB 程度を対象としてクラス間の関係抽出を行った．Common Crawl コーパスを利用してクラス間の関係抽出を行うため，Amazon Elastic MapReduce(注7)（Hadoop クラスタ）の環境を用いて解析を行った．実際には，3 章で説明した手法のうち，トライ辞書による語句間の関係抽出を Amazon Elastic MapReduce を用いて並列処理し，以降の処理は 1 台のローカルマシンで行った．語句間に出現する単語数 K は 3 とした．

5.1　処 理 時 間

約 15TB のテキストに対し，Hadoop クラスタを構成して関係抽出を行った．使用した計算機は，メモリが 7 GiB（ギビバイト），CPU は 20 ECU（1ECU は 1.0-1.2 GHz 2007 Opteron または 2007 Xeon プロセッサの CPU 能力と同等の性能を持つ），プラットフォームは 64 ビットで，この計算機を 100 台用いて解析を行った．このとき，語句間の関係抽出にかかった時間は 14 時間 8 分であった．このことから，テラバイトスケールのテキストから高速に語句間の関係を抽出できていることがわかる．コーパスサイズが大きく，同じテキストに対して他の手法と処理時間の比較を行うことが困難であるため，小規模なテキストデータに対して，先行研究 [21] で用いた形態素解析による語句間の関係抽出手法と比較を行った．表 1 は Reuter Corpus RCV1(注8) の 1997 年 6 月 1 日から 1997 年 8 月 19 日までの記事に対し，高速なモデルによる形態素解析（Stanford POS Tagger [19], english-left3words-distsim モデル）を用いた手法とトライ辞書のみを用いる提案手法によって語句間の関係を抽出したときの処理時間を計測した結果である．トライ辞書のみを用いた手法は，形態素解析を用いた手法と比較して処理時間が 4 分の 1 以下に抑えられていることがわかる．大規模な Web コーパスにおいても処理時間の比が変化しないことを考慮すると，提案手法の処理速度の速さは大きな利点として機能しているといえる．

5.2　抽出した関係の規模

約 15TB のテキストから抽出した関係の数を表 2 に示す．提案手法により得られたクラス間の関係は全部で 1 億 7 千万種類となり，先行研究や PATTY [15] と比較してもその規模の大きさがわかる．特に，PATTY では提案手法と同様に WordNet のクラスを用いていることから，規模の観点では，得られたクラス間の関係に十分価値があるといえる．先行研究や PATTY では，Wikipedia の全テキストを対象としてクラス間の関係抽出を行っているが，これは，テラバイトスケールのテキストに対して手法を適用すると処理時間が膨大となることに起因している．本研究では，文字列処理のみによる高速な関係抽出により，大規模なテキストデータへの適用を可能としている．その結果，大規模な数のクラス間の関係を取得できている．

(注7)：http://aws.amazon.com/jp/elasticmapreduce/
(注8)：http://trec.nist.gov/data/reuters/reuters.html

表 3　Web コーパスから抽出したクラス間の関係の精度

本研究	0.86
先行研究 [21]	0.96
PATTY [15]（判定基準は異なる）	0.85

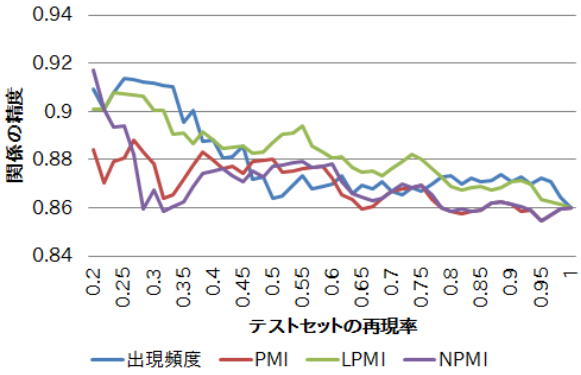

図 3　テストセットの再現率とエラー率の関係

5.3　抽出した関係の精度

抽出した関係の精度について評価するため，被験者によるサンプルの判定を行った．ドメイン非依存な事実関係の抽出に関する研究 [16] や先行研究 [21] で用いられているクラスから 20 種類を基準として選び，ランダムにそれぞれ 30 件ずつ，計 600 の関係をサンプルとして抽出し，テストセットを作成した．被験者には，クラス間の関係が正しいかどうかの判定，および，正しい場合にはその関係がどの程度明確かを 1（曖昧である）から 5（明確である）の 5 段階で判定させた．たとえば，「person and university」という関係は誤りではないがあまり情報を持っていない曖昧な関係であり，「person studied at university」という関係は明確な意味を持つ．具体的には，3 人の被験者のうち，2 人以上が正しいと判定した場合に，その関係が正しいと判断した．また，正しいと判断された関係について，被験者による明確さの判定の中央値（一人の被験者が誤っていると判断した関係に対しては残り二人によるスコアの下限値）を算出し，出現頻度，PMI，LPMI，および NPMI との関係を調査した．

表 3 にテストセット全体の精度，図 3 に頻度，PMI，LPMI，NPMI それぞれのスコアでソートしたときのテストセットの再現率（カバー率）と精度との関係を示す．なお，再現率が 0.2 以下の精度については，関係数が 120 以下と少なく，信頼できる値が算出できないため省いた．まず，全体でみると，本研究で抽出したクラス間の関係の精度は先行研究に劣っている．これは，本研究の提案手法では形態素解析を用いずに文字列処理のみで関係を抽出しているためであると考えられる．一方，図 3 より，出現頻度や PMI，LPMI，NPMI のスコアが上位の関係のみをみると，精度が向上する傾向にあることがわかる．中でも，LPMI あるいは出現頻度によるスコアでソートしたときに，全般的に精度が高くなっている．

図 4 は頻度，PMI，LPMI，NPMI それぞれのスコアでソー

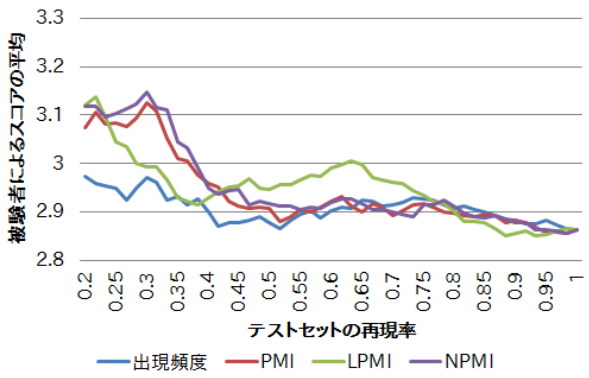

図 4　テストセットの再現率と被験者による判定スコアの平均の関係

トしたときのテストセットの再現率（カバー率）と被験者による判定スコアの平均との関係を表している．判定スコアの平均が高いほど，より明確な関係が多く含まれていることを意味している．図 4 より，出現頻度以外のスコアにおいて上位の関係ほど，明確な関係が多くなる傾向があることがわかる．出現頻度の高い関係がそれほど明確でないことの理由として，一般的すぎるクラス間の関係が多いことが挙げられる．このようなクラス間の関係は誤りではないため単純な正誤の精度としては高くなるが，関係の明確さという指標でみると出現頻度はスコアリング手法としてあまり適していないといえる．一方で，PMI，LPMI，NPMI では，クラスと関係パターンの関連度を考慮しているため，あまり出現頻度が多くなくても，明確な関係に対して高いスコアを付与できる．図 3 の結果を踏まえると，LPMI が他の手法に比べて有効であると考えられる．

以上の結果から，提案手法によって得られたクラス間の関係は，全体としては精度の点で従来手法に劣っているが，確率的な関係のスコアリングによって上位の関係のみをみることで，精度の問題を解決できる可能性がある．

5.4　抽出したクラス間の関係の例

抽出したクラス間の関係の中から，関係パターンと片側のクラスをクエリとし，LPMI のスコア順に関係を 3 つ取得した例を表 4 に示す．太字はクエリによって得られたクラスを表している．表 4 より，各クエリに対して，やや不適切な関係も一部みられるが，多くの場合，適当な粒度でクラスが取得できていることがわかる．たとえば，本論文の背景では「テニス選手がテニス選手を破る」という例を紹介したが，「tennis player beat X（テニス選手が X を破る）」というクエリに対して，出現頻度や PMI などのスコアが最も高いクラスが「tennis player」となっている．また，「X acquired company」というクエリに対し，「company」というクラスの PMI，LPMI，NPMI の各スコアが最も高くなっている．この例では，「company」よりも出現頻度が多い「institution」や「organization」は，クラス単体での出現頻度が大きいため，PMI，LPMI，NPMI の各スコアが低くなり，結果として「company」という適当な粒度でクラスを取得できている．

「wrote」という関係パターンに注目すると，「person wrote X」，「musician wrote X」，「lawyer wrote X」という各クエリに対して，それぞれに適したクラスが得られている．これは，同じ「wrote」という関係パターンでも，伴って出現するクラスによって意味合いが異なることを表している．一般的に「人が X を書いた」といえば，X には本や小説などが入ると考えられる．また，「ミュージシャンが X を書いた」の場合，X は曲である可能性が高い．さらに，「法律家が X を書いた」という場合では，X は単なる書き物ではなく，法律という専門的な書き物であると予測できる．このように，X に関する情報を持っていなくても，X がどういうものかということを，学習したクラス間の関係を用いることで推測できる．

6.　おわりに

本研究では，Wikipedia から得られた語句，エンティティ，および WordNet のクラスを用いて，テラバイトスケールの Web コーパスからクラス間の関係を抽出する手法を提案した．提案手法では，Web コーパスから語句を抽出し，2 つの語句が近接して出現する箇所から語句間の関係を抽出する．そして，得られた語句間の関係から，Wikipedia のエンティティを介して語句を WordNet のクラスに置き換えることで，クラス間の関係を取得する．また，抽出した関係に対して，出現確率にもとづくスコアリングを行い，関係の精度向上を図る手法を提案した．実際に 10 億以上の Web ページ（約 15TB）から抽出した関係の規模や精度，そのときの処理時間について評価を行い，提案手法およびそれによって抽出したクラス間の関係の有用性を確認した．

今後の課題として，今回の手法で使用しなかった曖昧性のある語句の利用や，抽出した関係に対しての自然言語処理技術の適用が挙げられる．提案手法では，簡単な文字列処理によって抽出可能な関係のみに焦点を当てて関係抽出を行ったが，今後はより高度な処理を，処理速度を考慮しつつ取り入れる予定である．また，同じ意味を表す関係の集約，クラス間の関係の多言語化を検討している．提案手法によって抽出した関係は文字列によって表現されているため，これを意味レベルに変換，すなわち，同じ意味を持つ関係を 1 つにまとめて定義することで，より汎用性のある知識として利用できる．また，クラス間の関係の多言語化についても，同じ意味を持つ関係を，言語の枠を超えて定義することにより実現できると考えられる．

謝辞　本研究の一部は，日本学術振興会特別研究員奨励費 (24-807) の助成によるものである．ここに記して謝意を表す．

文　献

[1] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, “Open Information Extraction from the Web,” Proc. of IJCAI, pp.2670–2676, 2007.
[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge,” Proc. of SIGMOD, pp.1247–1249, 2008.
[3] D.T. Bollegala, Y. Matsuo, and M. Ishizuka, “Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web,” Proc. of WWW, pp.151–160, 2010.
[4] G. Bouma, “Normalized (Pointwise) Mutual Information in Collocation Extraction,” Proc. of GSCL, pp.31–40, 2009.
[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R.

表 4　抽出したクラス間の関係の例

クエリ	クラス間の関係	出現頻度	PMI	LPMI	NPMI
X acquired company	**company** acquired company	650	9.1145	5924.3992	0.2052
	institution acquired company	703	8.2089	5770.8676	0.1858
	organization acquired company	703	7.1760	5044.7589	0.1624
institution is based on X	institution is based in **region**	302	4.0894	1234.9977	0.0877
	institution is based in **location**	302	3.9204	1183.9725	0.0841
	institution is based in **district**	271	4.1120	1114.3511	0.0876
person graduated from X	person graduated from **educational institution**	65	6.1243	398.0786	0.1199
	person graduated from **body**	64	5.4762	350.4746	0.1072
	person graduated from **institution**	65	2.9981	194.8778	0.0587
X invented electronic device	**pioneer** invented electronic device	3	6.2593	18.7779	0.1044
	originator invented electronic device	3	6.2591	18.7774	0.1044
	interior designer invented electronic device	3	6.2084	18.6253	0.1036
company was founded by X	company was founded by **executive**	20	6.9602	139.2037	0.1278
	company was founded by **administrator**	20	6.4629	129.2584	0.1187
	company was founded by **head**	20	6.1806	123.6125	0.1135
country attacked X	country attacked **ethnic group**	5	6.5849	32.9243	0.1126
	country attacked **state**	5	5.2365	26.1823	0.0896
	country attacked **political unit**	5	5.2347	26.1733	0.0895
X adjacent to urban area	**location** adjacent to urban area	9	2.6848	24.1630	0.0473
	highway adjacent to urban area	2	8.6378	17.2757	0.1414
	region adjacent to urban area	7	2.4553	17.1872	0.0427
tennis player beat X	tennis player beat **tennis player**	193	15.0577	2906.1371	0.3142
	tennis player beat **champion**	177	12.6894	2246.0303	0.2634
	tennis player beat **rival**	177	12.5984	2229.9152	0.2615
person wrote X	person wrote **novel**	7	5.0326	35.2281	0.0875
	person wrote **fiction**	7	4.8097	33.6676	0.0837
	person wrote **literary composition**	7	4.7584	33.3087	0.0828
musician wrote X	musician wrote **music**	67	8.1238	544.2951	0.1594
	musician wrote **auditory communication**	67	8.0711	540.7637	0.1583
	musician wrote **song**	48	9.1085	437.2096	0.1754
lawyer wrote X	lawyer wrote **law**	2	10.3136	20.6272	0.1688
	lawyer wrote **fundamental law**	2	10.3136	20.6272	0.1688
	lawyer wrote **statesman**	3	4.8073	14.4220	0.0802

Hruschka, and T.M. Mitchell, “Toward an Architecture for Never-Ending Language Learning,” Proc. of AAAI, pp.1306–1313, 2010.

[6] O. Etzioni, M. Cafarella, D. Downey, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates, “Unsupervised Named-Entity Extraction from the Web: An Experimental Study,” Artificial Intelligence, vol.165, no.1, pp.91–134, 2005.

[7] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam, “Open Information Extraction: The Second Generation,” Proc. of IJCAI, pp.3–10, 2011.

[8] A. Fader, S. Soderland, and O. Etzioni, “Identifying Relations for Open Information Extraction,” Proc. of EMNLP, pp.1535–1545, 2011.

[9] C. Fellbaum, WordNet: An Electronic Lexical Database, The MIT Press, 1998.

[10] H.H. Hoang, S.N. Kim, and M.Y. Kan, “A Re-examination of Lexical Association Measures,” Proc. of MWE, pp.31–39, 2009.

[11] J. Hoffart, F.M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum, “YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages,” Proc. of WWW, pp.229–232, 2011.

[12] D. Milne and I.H. Witten, “Learning to Link with Wikipedia,” Proc. of CIKM, pp.509–518, 2008.

[13] T.P. Mohamed, E.R.H. Jr., and T.M. Mitchell, “Discovering Relations between Noun Categories,” Proc. of EMNLP, pp.1447–1455, 2011.

[14] G.L. Murphy, The Big Book of Concepts, The MIT Press, 2002.

[15] N. Nakashole, G. Weikum, and F. Suchanek, “PATTY: A Taxonomy of Relational Patterns with Semantic Types,” Proc. of EMNLP, pp.1135–1145, 2012.

[16] M. Paşca, “Organizing and Searching the World Wide Web of Facts - Step Two: Harnessing the Wisdom of the Crowds,” Proc. of WWW, pp.101–110, 2007.

[17] E. Riloff and R. Jones, “Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping,” Proc. of AAAI, pp.474–479, 1999.

[18] F.M. Suchanek, G. Kasneci, and G. Weikum, “YAGO: A Core of Semantic Knowledge,” Proc. of WWW, pp.697–706, 2007.

[19] K. Toutanova, D. Klein, C.D. Manning, and Y. Singer, “Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network,” Proc. of HLT-NAACL, pp.252–259, 2003.

[20] F. Wu and D.S. Weld, “Open Information Extraction using Wikipedia,” Proc. of ACL, pp.118–127, 2010.

[21] 白川真澄, 中山浩太郎, 荒牧英治, 原 隆浩, 西尾章治郎, “Wikipedia と Freebase の知識を利用したテキストからの上位概念間の関係抽出,” WebDB Forum, 2012.