DBSJ Journal, Vol.12, No.1
June 2013

Wikipedia と WordNet を利用した膨大なテキストコーパスからのクラス間関係抽出

Extracting Relations Between Classes from Huge Text Corpus Using Wikipedia and WordNet

白川 真澄♠　中山 浩太郎♥
荒牧 英治♦　原 隆浩♠　西尾 章治郎♠

Masumi SHIRAKAWA　Kotaro NAKAYAMA
Eiji ARAMAKI　Takahiro HARA　Shojiro NISHIO

本論文では，基盤知識および文字列処理技術を用いて膨大な Web テキスト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．提案手法は，人がクラス間の関係を学習する方法を模倣し，テキストから語句間の関係を抽出した後，Wikipedia の記事を介して語句を WordNet のクラスに置き換えることでクラス間の関係を学習する．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．

In this paper, we propose a domain-independent method of extracting relations between classes (class relations) from huge Web text corpus using knowledge bases and string processing techniques. The proposed method imitates human way of learning class relations, i.e., extracting relations between terms and replacing the terms by WordNet classes through Wikipedia articles. We extracted more than one hundred million class relations from over a billion Web pages (approx. 15TB). Evaluation results revealed the efficiency of our method and the availability of extracted class relations.

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（クラス）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[8]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．

本研究では，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，膨大なWeb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する手法を提案する．提案手法では，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，Wikipedia[1]とWordNetの知識を用いて語句をクラスに置き換えてから関係を抽出する．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，文字列処理技術のみを適用する．また，関係の出現頻度をもとに確率的なスコアを与える手法により，より明確な意味を持つ関係への効率的なアクセスが可能となる．実際に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，そのときの処理時間および抽出した関係の規模や精度について考察する．

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 [4]やBankoらのTextRunner [1]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．関係パターン（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処理を反復させることにより，関係（例：Tokyo, capitalOf, Japanの3つ組）の数を拡大していく．BollegalaらのRelational Duality [3]では，同じ種類の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．これらの関係抽出手法では，1）いかに多くの事実関係を 2）

♠ 正会員　大阪大学大学院情報科学研究科
{shirakawa.masumi,hara,nishio}@ist.osaka-u.ac.jp
♥ 正会員　東京大学知の構造化センター
nakayama@cks.u-tokyo.ac.jp
♦ 非会員　京都大学デザイン学ユニット
eiji.aramaki@gmail.com

[1] http://www.wikipedia.org/

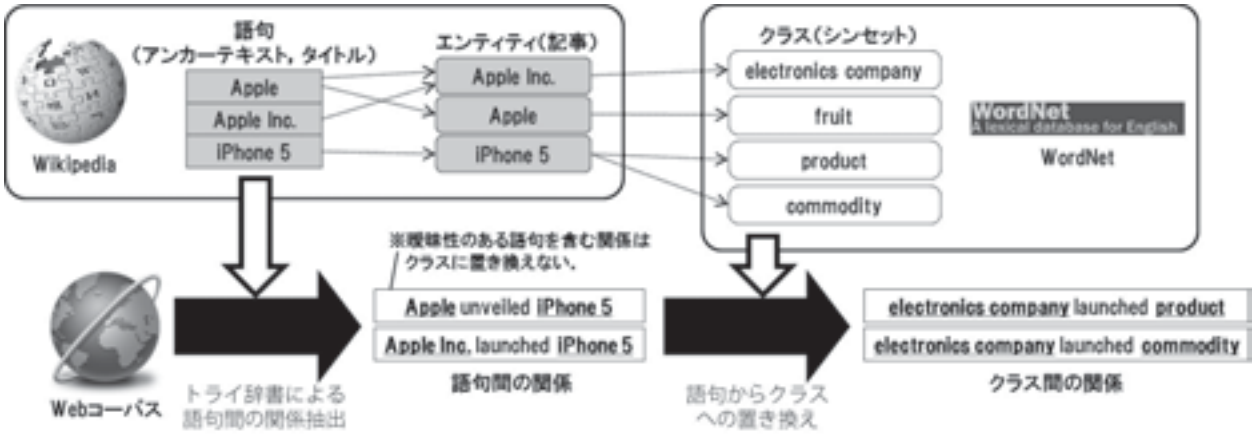

図 1　提案手法の概要
Fig.1　Overview of the proposed method.

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，語句間やエンティティ間の関係ではなく，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，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．ReVerb [5]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約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 [9]では，YAGO [11]を用いてWikipediaおよびWordNet [6]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，膨大なテキストへの適用を可能としている．

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．まず，人がどのように概念（クラス）を利用して関係を学習・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．また，次章では，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

以下では，人がどのように概念間の関係を学習し推測するかについて説明する．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．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くても，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徐々に学習していく．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と概念レベルでの認識が関係の学習において重要な役割を果たしている．コンピュータにおいても，このように概念レベルで関係を学習させることにより，人が行うような未知語を含むテキストの意味推測が可能になると考えられる．

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1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を関係パターンとして取得する．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1の各処理について説明する．

3.3 使用する前提知識

提案手法ではWikipedia，WordNet [6]，およびYAGO [11]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5月時点において，最も多い英語版で422万，日本語版で85万である．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いることから，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（シンセット（synset）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

Wikipedia のアンカーテキストと記事のタイトルから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，本研究では英語を対象としているため，英単語と特定の記号（「'」「,」「-」）のみから成る関係パターンのみを抽出する．英単語リストには 12Dicts[2]を用いている．

Wikipedia のアンカーテキストと記事のタイトル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）[3]」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つなげることで，Wikipedia を整理する役目を持っている．そのため，アンカーテキスト（およびタイトル）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．

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，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つながっている．そこで，ある語句をみたとき，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．

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

$CountLinks(t, e)$ は t から e へのリンク数である．なお，記事のタイトルは，その記事へのアンカーテキストの 1 つとしてカウントする．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，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．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，人が学習するクラス間の関係により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．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）［7］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）出現しているかを数値化できる．

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

[2] http://wordlist.sourceforge.net/

[3] http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Wikify

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．

$$PMI(c,r) = \log_2 \frac{P(c,r)}{P(c)P(r)}$$

$P(c, r)$ 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．

$$PMI(c_L,r,c_R) = \log_2 \frac{P(c_L,r,c_R)}{P(c_L)P(r)P(c_R)}$$

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という問題がある．これを解決する手法の 1 つとして，LPMI 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，

$$LPMI(c,r) = PMI(c,r) \cdot Freq(c,r)$$

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．

$$LPMI(c_L,r,c_R) = PMI(c_L,r,c_R) \cdot Freq(c_L,r,c_R)$$

5. 10 億ページからの関係抽出

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．

解析対象とするコーパスとして，Common Crawl コーパス[4]を利用した．Common Crawl コーパスは，Amazon Web Services で使用できる Web コーパスであり，全サイズは 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[5]（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の最大値 K は 3 とした．

表 1　小規模なテキストデータに対する処理時間
Table 1　Computational time for the small dataset.

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

表 2　抽出したクラス間の関係数
Table 2　Number of extracted relations between classes.

PATTY [9]	350,569
本研究	**170,701,978**

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，形態素解析を用いた場合と比較を行った．表 1 は RCV1 コーパス[6]の 1997 年 6 月 1 日から 1997 年 8 月 19 日までの記事に対し，処理速度を重視したモデルによる形態素解析（Stanford POS Tagger [12], 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クラス間の関係数を表 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約 1 億 7 千万種類となった．PATTY [9]でも主に提案手法と同じ 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．PATTY では，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ため，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．一方，本研究では，文字列処理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，既存研究と比べてマイナーな関係を数多く網羅でき，膨大な数のクラス間の関係を取得できている．

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 [10]で用いられているクラスの中から 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

4 http://commoncrawl.org/

5 http://aws.amazon.com/jp/elasticmapreduce/

6 http://trec.nist.gov/data/reuters/reuters.html

表 3　抽出したクラス間の関係の精度（判定基準は異なる）
Table 3　Accuracy of extracted relations between classes (evaluation criteria may be different).

PATTY [9]	0.85
本研究	0.86

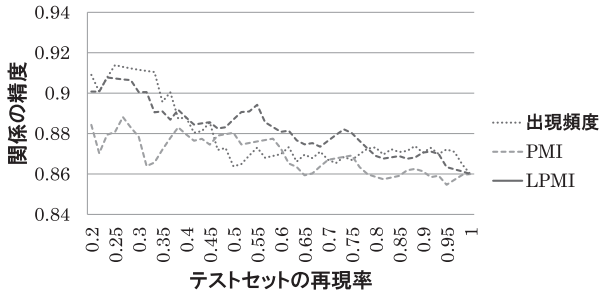

図 2　テストセットの再現率と関係の精度
Fig.2　Recall of the test set and accuracy of relations.

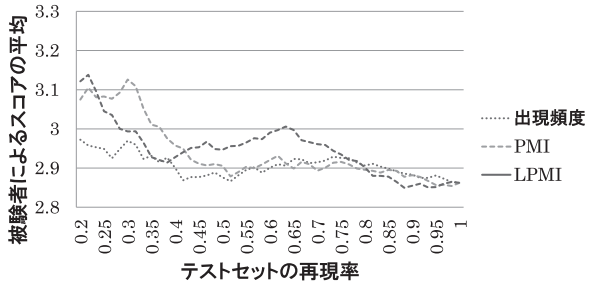

図 3　テストセットの再現率と被験者によるスコアの平均
Fig.3　Recall of the test set and average score by examinees.

ある）から 5（明確である）の 5 段階で判定させた．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 との関係をそれぞれ調査した．

表 3 にテストセット全体の精度，図 2 に出現頻度，PMI，LPMI 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は，PATTY と同等の精度となっている（これらの精度は異なる被験者らによるものであるため純粋な比較ではないことに注意）．また，図 2 より，出現頻度や PMI，LPMI 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 3 は出現頻度，PMI，LPMI 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 3 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あまり明確でないことの理由として，一般的なクラスによる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 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，出現頻度が少なくても，明確な関係に対して高いスコアを付与できる．図 2 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．

このようなスコアリング手法により，関係数の増大に伴って低下する精度の問題に対し，正しい関係やより明確な関係に効率的にアクセスできる．したがって，実用的にはランダムに抽出したサンプルによる精度よりも高い精度を実現可能であるといえる．

5.4　抽出した関係の例

実際にクラス間の関係を利用する場面を想定し，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て，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，多くの場合，各クエリに対して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，「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. まとめと今後の課題

本研究では，Wikipedia と WordNet を用いてテラバイトスケールの Web テキスト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．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．膨大な Web テキストコーパスから得られたクラス間の関係は，その関係の種類の数において既存研究に大きく勝っており，さらに適切なスコアリング手法を用いることで，より高い精度の達成や，明確な意味を持つ関係への効率的なアクセスが可能となる．

今後の課題として，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，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]
本研究の一部は文部科学省国家課題対応型研究開発推進事業 －次世代 IT 基盤構築のための研究開発－「社会システ

表 4　抽出したクラス間の関係の例
Table 4　Examples of extracted relations between classes.

クエリ	クラス間の関係	出現頻度	PMI	LPMI
X acquired company	**company** acquired company	650	9.11	5924.40
	institution acquired company	703	8.21	5770.87
	organization acquired company	703	7.18	5044.76
person graduated from X	person graduated from **educational institution**	65	6.12	398.08
	person graduated from **body**	64	5.48	350.47
	person graduated from **institution**	65	3.00	194.88
tennis player beat X	tennis player beat **tennis player**	193	15.06	2906.14
	tennis player beat **champion**	177	12.69	2246.03
	tennis player beat **rival**	177	12.60	2229.92
person wrote X	person wrote **novel**	7	5.03	35.23
	person wrote **fiction**	7	4.81	33.67
	person wrote **literary composition**	7	4.76	33.31
musician wrote X	musician wrote **music**	67	8.12	544.30
	musician wrote **auditory communication**	67	8.07	540.76
	musician wrote **song**	48	9.11	437.21
lawyer wrote X	lawyer wrote **law**	2	10.31	20.63
	lawyer wrote **fundamental law**	2	10.31	20.63
	lawyer wrote **statesman**	3	4.81	14.42

ム・サービスの最適化のための IT 統合システムの構築」(2012 年度～2016 年度) の助成による.

[文献]

[1] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M. and Etzioni, O.: Open Information Extraction from the Web, Proc. of IJCAI, pp. 2670–2676 (2007).
[2] Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J.: Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge, Proc. Of SIGMOD, pp. 1247–1249 (2008).
[3] Bollegala, D. T., Matsuo, Y. and Ishizuka, M.: Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web, Proc. of WWW, pp. 151–160 (2010).
[4] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. and Yates, A.: Unsupervised Named-Entity Extraction from the Web: An Experimental Study, Artificial Intelligence, Vol. 165, No. 1, pp. 91–134 (2005).
[5] Fader, A., Soderland, S. and Etzioni, O.: Identifying Relations for Open Information Extraction, Proc. Of EMNLP, pp. 1535–1545 (2011).
[6] Fellbaum, C.: WordNet: An Electronic Lexical Database, The MIT Press (1998).
[7] Hoang, H. H., Kim, S. N. and Kan, M.-Y.: A Re-examination of Lexical Association Measures, Proc. of Workshop on Multiword Expressions, ACL-IJCNLP, pp. 31–39 (2009).
[8] Murphy, G. L.: The Big Book of Concepts, The MIT Press (2002).
[9] Nakashole, N.,Weikum, G. and Suchanek, F.: PATTY: A Taxonomy of Relational Patterns with Semantic Types, Proc. of EMNLP, pp. 1135–1145 (2012).
[10] Pasca, M.: Organizing and Searching the World Wide Web of Facts - Step Two: Harnessing the Wisdom of the Crowds, Proc. of WWW, pp. 101–110 (2007).
[11] Suchanek, F. M., Kasneci, G. and Weikum, G.: YAGO: A Core of Semantic Knowledge, Proc. of WWW, pp. 697–706 (2007).
[12] Toutanova, K., Klein, D., Manning, C. D. and Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network, Proc. of HLT-NAACL, pp. 252–259 (2003).

白川 真澄　Masumi SHIRAKAWA
大阪大学大学院情報科学研究科特任助教. 2013 年大阪大学大学院情報科学研究科博士後期課程修了，博士（情報科学）. Web マイニングに関する研究に従事. 情報処理学会，自然言語処理学会各会員.

中山 浩太郎　Kotaro NAKAYAMA
東京大学知の構造化センター特任講師. 2007 年大阪大学大学院情報科学研究科博士後期課程修了，博士（情報科学）. 同年 4 月から大阪大学大学院情報科学研究科特任研究員. 人工知能，Web マイニングに関する研究に従事. ACM，情報処理学会，人工知能学会各会員.

荒牧 英治　Eiji ARAMAKI
京都大学デザイン学ユニット特定准教授. 2005 年東京大学大学院情報学研究科博士後期課程修了，博士（情報理工学）. 自然言語処理，医療情報の研究に従事. 情報処理学会，言語処理学会，医療情報学会，ACL 各会員.

原 隆浩　Takahiro HARA
大阪大学大学院情報科学研究科准教授. 1997 年大阪大学大学院工学研究科博士前期課程修了，工学博士. データベースシステム，分散処理の研究に従事. IEEE，ACM，情報処理学会，電子情報通信学会各会員.

西尾 章治郎　Shojiro NISHIO
大阪大学大学院情報科学研究科教授，サイバーメディアセンター長. 1975 年京都大学工学部卒業. 1980 年同大学院工学研究科博士後期課程修了，工学博士. 京都大学工学部助手等を経て，1992 年大阪大学工学部教授となり，現職に至る. 文部科学省科学官，大阪大学理事・副学長等を歴任. データ工学の研究に従事. 本会理事，監事を歴任し，現在，会長を務める. 紫綬褒章を受章し，本会より功労賞，論文賞を受賞.