B2IM2024

修士論文

大規模 Web データからの関係知識の獲得

高瀬翔

2014 年 2 月 28 日

東北大学 大学院
情報科学研究科 システム情報科学専攻

本論文は東北大学 大学院情報科学研究科 システム情報科学専攻に
修士 (工学) 授与の要件として提出した修士論文である。

高瀬翔

審査委員：

乾 健太郎 教授　　（主指導教員）
大町 真一郎 教授
木下 哲男 教授
岡崎　直観 准教授　（副指導教員）

大規模 Web データからの関係知識の獲得 *

高瀬翔

内容梗概

「東北大学は仙台に存在する」や「ガルシア・マルケスは百年の孤独の著者である」のように，何らかの意味的関係を持つ名詞対を関係インスタンスと呼ぶ．関係インスタンスは質問応答や推論など，自然言語処理の様々な応用に用いられる，非常に重要な知識であり，近年，大規模な Web 文書からありとあらゆる関係インスタンスを収集しようという研究が盛んに行われている．しかしながら，既存の研究では，同一の関係をまとめあげることをないがしろにしており，また，関係インスタンスを抽出する際に，述語を用いない表現を無視している．後者は，「X の Y」や「X による Y」のような，広く使われる表現を使用できないことを意味しており，これにより，Web のロングテールを扱えていない事が推測される．本研究では，日本語 Web 文書から，関係のありそうな名詞対を簡単なヒューリスティクで収集し，それを用いて，関係パタンを広範に収集する．この関係パタンをまとめあげることで，既存の研究よりも遥かに多くのインスタンスを収集できるような関係パタンの知識を構築する事を目的とする．

キーワード

自然言語処理，情報抽出，知識獲得，分散並列処理

*東北大学 大学院情報科学研究科 システム情報科学専攻 修士論文, B2IM2024, 2014 年 2 月 28 日.

目 次

図 目 次

表 目 次

1　はじめに

ビッグデータという言葉に見られるように，Web 上には人々が日々創出するデータ，とりわけ，テキストデータが大量に存在する．これら大量のテキストデータを言語資源（コーパス）とし，ここから単語の意味や単語間の意味的関係などの知識を構築することは，情報検索や推論，質問応答など自然言語処理の応用分野のために必要不可欠である [1, 2, 3]．例えば意味的関係として，**上位**｛ノーベル賞作家，ガルシア・マルケス｝，**著者**｛ガルシア・マルケス，百年の孤独｝という知識を持っていたとすると,「百年の孤独で有名なノーベル賞作家は誰か？」という問いに対し，適切な推論によって,「ガルシア・マルケス」と答えることができる．この「上位」や「著者」は意味的関係の種類を表しており,｛ノーベル賞作家，ガルシア・マルケス｝や｛ガルシア・マルケス，百年の孤独｝のように，特定の意味的関係にある単語対を関係インスタンスと呼ぶ．関係インスタンスをコーパスから自動的に獲得する手法については様々な研究が為されてきた．

伝統的に，関係インスタンスの獲得は，獲得対象である特定の関係について，人手で事前知識を作成し，入力とする必要がある．例として，対象の関係に属するインスタンスについて書かれた文とそうでない文をあらかじめ人手で分類しておき，両者の比較から関係インスタンスを表す表現（関係パタン）を学習し，この関係パタンを用いて新たな関係インスタンスを得る手法 [4] や，対象の関係に属する少数のインスタンス集合（あるいは関係パタンの集合）から，関係パタン（あるいは関係インスタンス）を獲得し，これを元に再び関係インスタンスを獲得するというように，関係インスタンスと関係パタンを交互に獲得することで，大規模な関係インスタンス集合を得る手法 [5, 6, 7] などがある．これらの手法では，新たな関係インスタンスを正確に識別，獲得するために適切な事前知識の入力が必要であるが，このような事前知識の作成は，しばしば専門的な知見が必要となる．また，獲得できる関係インスタンスが，あらかじめ定めた特定の関係に属するものに限られるため，複数の関係についての知識を獲得したい場合，それに応じた前提知識の作成が必要となる．

柔軟な推論システムや，どのような質問にも応じられる質問応答システム構築のためには，ありとあらゆる関係インスタンスを獲得しておく必要がある．あらゆ

る関係インスタンス獲得のために，人手で前提知識を作成することは途方もない労力が必要であり，また，仮に多様な関係を収集するための前提知識を作成したとしても，Web 上に記述されている，予期していない関係については取りこぼしてしまう可能性がある．これを解決するために，近年，*Open Information Extraction*（Open IE）システムの開発が盛んに研究されている [3, 8, 9, 10, 11, 12]．Open IE システムとは，人手での事前知識を必要とせず，テキストに書かれているあらゆる関係インスタンスの獲得を自動で行うシステムである [8, 12]．Open IE システムは，関係を表す表現は一般的に動詞や動詞句であるというような，統語的な特徴に着目し，関係パタンの特定と関係インスタンスの抽出を行う [9]．例えば,「1967 年，ガルシア・マルケスは百年の孤独を発表した」という文について，Open IE システムは,「発表する」という動詞[1] で表される関係について言及していると判断し，**発表する**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを得る．

Open IE システムは大規模なテキストコーパスから，大量の関係インスタンスを獲得することに成功しているが [9, 10]，同一の関係をまとめられていないという問題がある．例えば,「1967 年，ガルシア・マルケスは百年の孤独を発表した」という文と「ガルシア・マルケスは 18ヶ月かけて百年の孤独を執筆した」という文について，人間ならば，どちらの文にも「ガルシア・マルケスは百年の孤独の著者である」という事柄が書いてあることが分かる，すなわち，**著者**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを獲得できる．しかしながら，既存の Open IE システムでは，それぞれの文から，**発表する**｛ガルシア・マルケス，百年の孤独｝，**執筆する**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを獲得するのみであり，この二つの関係インスタンスがどちらも，著者｛ガルシア・マルケス，百年の孤独｝という関係インスタンスと同一であると判断することはできない．その結果，本来ならば同一の関係インスタンスを，様々な関係に属する別々のインスタンスとして獲得してしまい，構築される知識が煩雑なものとなってしまう．

また，Open IE システムは，関係パタンを動詞や動詞句に限っているため，関

[1] 日本語において，正確には「発表」というサ変名詞と「する」という動詞に分割されるが，ここでは簡単のため，このように記す．

係を表している表現の多くを認識できない．例えば,「この焼酎は，ガルシア・マルケスの小説である百年の孤独にちなんでいる」という文から，人間であれば，**著者**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを獲得できる．しかし，Open IE システムは「X の小説である Y」を関係パタンとして認識できないため，この文から，**著者**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを獲得することができない．関係インスタンスは動詞や動詞句以外で表されている場合も多いため [13]，コーパス上に書かれている，関係インスタンスを取りこぼしてしまう可能性がある．

本論文では，既存の Open IE システムのように，あらかじめ用意した知識を用いずに関係インスタンスを獲得することを目的とするのではなく，上記二点の問題を解決できるような，関係パタンの知識の構築を目的とする．提案手法では，最初に，何らかの関係を持つと推測される名詞対，すなわち，関係インスタンスと考えられる名詞対を簡単なヒューリスティクを用いて大量に収集する．次に，収集した名詞対を結ぶ表現を関係パタンとして抽出する．これにより,「X の小説である Y」はもちろん,「X の Y」や「X による Y」など非常に一般的な表現も関係パタンとして獲得することができる．次に，同一の関係を表す関係パタンをまとめあげるため，各関係パタンと共起している名詞対を元に，関係パタンのクラスタリングを行う．このクラスタリングによって,「X は Y を発表した」と「X は Y を執筆した」の二つの関係パタンが共に「X は Y の著者である」という関係を表すという知識を構築できる．この関係パタンの知識を用いて関係インスタンスを獲得することで,「1967 年，ガルシア・マルケスは百年の孤独を発表した」,「ガルシア・マルケスは 18ヶ月かけて百年の孤独を執筆した」という文から，**発表する**｛ガルシア・マルケス，百年の孤独｝，**執筆する**｛ガルシア・マルケス，百年の孤独｝という別々の関係インスタンスではなく，**著者**｛ガルシア・マルケス，百年の孤独｝という関係インスタンスを獲得することができる．

さらに，ここで構築した関係パタンの知識は，述語項構造解析など，深い意味処理のタスクへも有用であると考えられる．述語項構造解析とは，文中の，出来事などの事態を表す述語の意味解析と，その述語の取る意味的な項を同定するタスクである．例えば「1967 年，ガルシア・マルケスは百年の孤独を発表した」と

いう文について，述語「発表する」の意味的な項として,「ガルシア・マルケス」（動作主）,「百年の孤独」（対象）,「1967 年」（時間）を抽出する．また，ここでの述語「発表する」は執筆する，書く，という意味であると解析する．述語項構造解析における，述語に対する項と関係インスタンスの名詞対は同様である場合が多く [14]，さらに，述語の意味解析は関係の種類の特定と類似している．このように，述語項構造解析と Open IE は似ている点が多いため，大規模コーパスから獲得した関係パタンの知識を適用することで，述語項構造解析の精度を上げることも可能であると考えられる．

なお，本論文では，日本語 Web 文書をコーパスとし，関係パタンの抽出，クラスタリングを行う．Open IE システムの研究は，英語を対象にしたものが主流であり，それ以外の言語を扱っている研究はあまりない．特に，日本語の関係パタンの知識獲得については，DeSaeger らが公開しているもの [13] 以外には見られない．このため，日本語に応じた関係パタンの抽出手法を提案する事も本研究の貢献の一つである．

本論文での貢献は以下の点である．

- 大量の日本語 Web 文書をコーパスとして利用した，大規模な関係パタン知識の構築
- 「X による Y」のような，広く使われているが，述語で構成されない関係パタンの獲得
- 大量の Web 文書からの，大規模な関係パタンの知識獲得を実時間で実現する手法の提案

本論文の構成は以下のようになっている．2 節では，関連研究，特に，関係インスタンスの抽出手法についての研究について述べる．3 節では，大規模なデータの処理に必須な，分散並列処理の手法について述べる．4 節では提案手法，すなわち，関係パタンの知識の構築手法について述べる．5 節では提案手法の効果について，実験結果をもとに考察する．最後に 6 節において，この論文での結論を述べる．

2　関連研究

本節では，関連研究として，既存の関係抽出手法について，概要を紹介する．最初に，特定の関係に着目したインスタンス抽出として，人手でによる大量のアノテーションデータを利用した教師あり手法と，少数の事前知識を用いた半教師あり手法について述べる．次に，人手での事前知識なしに，Web 文書からあらゆる関係インスタンスの抽出を行う事を目的とする，Open Information Extraction の研究について述べる．次に，関係パタンの抽出やそのまとめあげなど，関係パタンの知識獲得を行った研究について述べる．最後に，クラスタリングの際に重要となる，高速な類似度計算手法について述べる．

2.1　特定の関係に着目したインスタンス抽出

2.1.1　教師あり手法

教師あり手法による関係抽出は，人手によってアノテーションされた言語資源の普及とともに発展してきた．教師あり手法による関係抽出は，まず第一に，人手でアノテーションされたデータから素性を抽出し，その素性によって，関係インスタンスをあらかじめ定義してある関係のいずれかに分類する分類器を学習し，学習したモデルを用いて新たに関係インスタンスを抽出する，というながれをとる．素性としては，係り受け関係（単語間の依存構造）のような統語的な構造や，単語の意味，品詞などが用いられる．Jiang らは人や国など単語の固有表現タイプや bag-of-words，文における単語連続や句構造解析，依存構造解析の結果など，様々な素性を設計し，組み合わせを変えて実験を行う事で，関係抽出にはどのような素性が有効かを調査した [15]．結果として，単語連続や句構造解析，依存構造解析の結果という統語的な素性については，組み合わせてもあまり性能が向上せず，句構造解析の結果を素性として用いれば十分である事を示した．分類器については，最大エントロピー法や SVM など，様々なものが用いられている [15, 16, 17]．

2.1.2 半教師あり手法

教師あり手法を利用するための，言語資源への人手でのアノテーションは，非常にコストが高い．また，関係の種類に特化した抽出手法を学習するため，様々な種類の関係や，異なるドメインに対し，柔軟に対応させる事が難しい．これを克服するため，人手で与える知識を減らした，半教師あり手法が研究されている．

半教師あり手法では，人手で用意した事前知識を元に，関係パタンと関係インスタンスの獲得を交互に繰り返し，インスタンス集合を獲得するという，Bootstrap 手法が広く利用されている [7, 18, 13, 19]．Bootstrap 手法では，まず事前知識として少数の関係インスタンスや関係パタンを人手で用意する．次に，この関係インスタンス（関係パタン）とコーパス中で相関の高い関係パタン（関係インスタンス）を獲得する．新たに関係パタン（関係インスタンス）を獲得したら，これを利用し，再びコーパスを読んで，相関の高い関係インスタンス（関係パタン）を獲得する．さらにまた，新たに獲得した関係インスタンスやパタンを使って，というように，交互にパタンとインスタンスの獲得を繰り返す．

Bootstrap 手法では，複数の関係について同時にインスタンス収集を行い，ある関係に属するインスタンスは別の関係に属さないなど，関係間の関係を利用することで精度を高める方法が提案されている [19, 18]．また，あらかじめ収集した知識を用いて，ある名詞対が関係インスタンスとなりうるかどうかを判定する手法も提案されている [13, 20]．DeSaeger らは，コーパス中の名詞をあらかじめクラスタリングして名詞のクラスを作成しておき，関係パタンにこの名詞クラスの情報を付与する手法を提案した [13]．彼らは関係インスタンスを収集する際に，関係パタンに付与されている名詞クラスに属さない名詞を除去することで，インスタンス獲得の精度を高められる事を示した．Carlson らは Bootstrap による名詞クラスの知識獲得と関係インスタンス獲得を同時に行う手法を提案した [20]．彼らは，DeSaeger らの手法のように，関係インスタンス獲得の際に名詞クラスに制限を適用し，精度向上を可能にした．

このように，人手の追加知識なしに，高精度での関係インスタンスを抽出する手法が研究されているが，Bootstrap 手法では，高い精度を達成できるような事前知識を選択する事が簡単ではないという問題もある [21]．

2.2 Open Information Extraction

上記のように，人手での知識を必要とする手法は，どのような手法でも知識を用意するためのコストが生じる．これに対し，人手での事前知識を一切必要とせず，Web 文書などのコーパス中から，ありとあらゆる関係インスタンスを抽出しようという研究が，近年盛んになってきている [3, 9, 12, 8, 10, 11, 10, 22]．このようなタスクを Open Information Extraction（Open IE）と呼ぶ．

Open IE では，動詞や動詞句のような，関係パタンを特定するための規則をあらかじめ記述しておき，その規則とマッチした表現及び，その項を関係インスタンスの候補として抽出する．さらに，あらかじめ小規模のデータセットで学習しておいた分類器を用いて，関係インスタンスの候補が実際に関係インスタンスであるかどうかを分類する [9, 12]．大規模な Web コーパスに対し，この処理を行う事で，大量の関係インスタンスの獲得に成功しており [9]，加えて，得られた大量の関係インスタンスはドメインに依存しない質問応答などにも有用な知識として用いられている [2]．

Open IE システム開発の初期には，関係パタンを特定するための規則は単語連続で表せる形，つまりは，正規表現で記述できるような単純な形式を用いていた [9] が，最新の研究では依存構造の利用など，統語的な特徴を活かした規則の記述も行われている [11]．また，DBpedia という知識リソースなどから自動的な手法で関係インスタンスである名詞対とそれを含む文を抽出し，これを訓練データとして，関係パタンや関係インスタンスを特定する規則を学習する手法も提案されている [22]．

しかしながら，既存の Open IE システムは，関係パタンとして着目している表現が動詞や動詞句のような述語および述語を含む表現に限られているため,「X ならば Y」や「X による Y」のような，述語を含まない表現を利用する事ができていない．また，Open IE システムの研究は，主に英語に対して行われているが，日本語のように語順の制約が緩い言語については，関係パタンを特定するために有効な規則が存在するかは疑わしい．DBpedia などの知識リソースが整備されている言語もそれほど多くはなく，Open IE システムの開発にはまだまだ多くの課題があると言える．

2.3 関係パタン知識の獲得

Open IE システムでは，動詞や動詞句のような関係パタンをそのまま関係の種類として用いるため,「執筆する」と「書く」のように同一の意味であっても異なる関係となってしまう．これに対処するために，得られた関係パタンのクラスタリングによるまとめあげが行われている [23, 24].

Lin らは，関係パタンが共起する名詞対にもとづいて関係パタン間の類似度を測定し，意味的に関連の深い関係パタンを判定する手法を提案した [25]．彼らは類似度を測定しているが，クラスタリングは行っておらず，また目的も，関係パタンのまとめあげではなく，推論規則の獲得としている．しかしながら，名詞対をもとに類似度を測定した場合,「X が Y に勝つ」と「X が Y に負ける」のような，意味的に逆の関係パタンの類似度が高くなってしまうこともあるという，類似度計算における本質的な難題を報告している．

Kok らは second-order Markov logic にもとづき，関係パタンと関係パタンの項となっている名詞をそれぞれクラスタリングする手法を提案した [26]．彼らの手法では，関係パタンや名詞はただひとつのクラスタにしか属する事ができず，複数の意味を持つ関係パタンを扱う事ができない．これに対し，Min らは関係パタンの項となっている名詞をクラスタリングして名詞のクラスを作成し，名詞のクラスによって関係パタンの意味を区別する手法を提案した [23]．関係パタンの表現自体が同じでも，異なる名詞クラスを項に持つ関係パタンはそれぞれ異なる関係パタンであるとし，複数の意味を持つ関係パタンの意味を分け，別々のクラスタに所属させるようにしている．

Nakashole らは関係パタンの同義表現のまとめあげに加え,「X は Y を上演する」は「X は Y を歌う」を包含するというような，関係パタン間の包含関係知識の構築手法を提案した [24]．彼らは YAGO という固有表現とそのクラスを記した知識資源を利用し，関係パタンに名詞のクラス情報を付与している．さらに，関係パタンに含まれる単語のうち，品詞やワイルドカードで置き換え可能なものは適宜置換する事で，より一般的な関係パタンを構築している．

近年の研究はどれも，大規模コーパスに対しても適用できるよう，スケーラビリティのある手法を謳っているが，実験においては，既存研究で得られた高信頼

度の関係インスタンスのみを対象としている [23] など，規模が小さい研究も存在する．また，どの研究も関係パタンは動詞や動詞句など，述語を含む表現のみを対象にしている．

2.4 高速な類似度計算手法

大規模データに対するクラスタリングを行う際には，データ間の類似度計算に必要とする時間が大きな問題となる．例えば，関係パタンと共起する名詞対を関係パタンの特徴ベクトルとして，関係パタンをクラスタリングすることを考える．このとき，パタンの特徴ベクトルの次元を k，パタンの数を n とすると，全パタンペアの類似度の計算には $O(kn^2)$ の時間を要する．これは，大規模に収集したパタン間（例えば $n = 1,000,000$）の類似度計算には，膨大な時間がかかってしまうことを意味する．一方で，同一のクラスタに属すであろう関係パタン，つまり，同一の関係を表すパタンはせいぜい 100 のオーダー程度しか存在しないであろうという直感がある．関係パタンをまとめあげるという目的では，この同一の関係を表すパタンを絞り込めていればよく，全パタン間の正確な類似度計算は必要ない．そこで，高速な近似近傍点探索手法により，類似度が高いと推測される関係パタンを絞り込み，その後，絞り込んだパタン間について，正確な類似度を計算し，それ以外のパタン間の類似度はゼロとして扱えば効率良く類似度の計算，クラスタリングが行える．

高速な近似近傍点探索手法として，Locality Sensitive Hashing（LSH）という手法が知られている．LSH とは，似た素性ベクトルを持つ要素が，高い確率で同じ値を取るようなハッシュ関数を利用し，近傍点を確率的に求めるアルゴリズムである．LSH では，元の要素間に対する距離尺度に応じたハッシュ関数を構築する必要がある．このハッシュ関数には，例えばコサイン類似度に対応するものとして，Charikar らによって提案されたものがある [27]．このハッシュ関数は，元の要素の素性ベクトル $\boldsymbol{u}$ に対し，同一次元数のランダムなベクトル $\boldsymbol{r}$ を用いて，

式 (1) のように定義される．

$$h_{\boldsymbol{r}}(\boldsymbol{u}) = \begin{cases} 1 & \text{if } \boldsymbol{r} \cdot \boldsymbol{u} \geq 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

このハッシュ関数では，元の要素の素性ベクトル u と v 間について，

$$\Pr[h_{\boldsymbol{r}}(\boldsymbol{u}) = h_{\boldsymbol{r}}(\boldsymbol{v})] = 1 - \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{\pi} \tag{2}$$

が成り立つ．

式 1 は言い換えれば，ランダムなベクトル $\boldsymbol{r}$ によって得られる $h_{\boldsymbol{r}}(\boldsymbol{u})$ と $h_{\boldsymbol{r}}(\boldsymbol{v})$ は，$\frac{\theta(\boldsymbol{u},\boldsymbol{v})}{\pi}$ の確率で異なることを示している．$\boldsymbol{u}$ と $\boldsymbol{v}$ のコサイン類似度が $cos(\theta(\boldsymbol{u}, \boldsymbol{v}))$ であることを考えると，d 個のランダムなベクトルと式 (1) から得られる d 次元のビットベクトルについて，$\boldsymbol{u}$ と $\boldsymbol{v}$ のコサイン類似度が β であるとすると，おおよそ $d \times \frac{arccos(\beta)}{\pi}$ ビットの違いが存在する．すなわち，d 次元のビットベクトルについて，ハミング距離を測る事で，元の要素間のコサイン類似度がしきい値以上の要素対を得る事ができる．このように，LSH を用いる事で，計算対象のベクトルの次元を，元々の次元 k よりも，遥かに小さい d に削減することができる．さらに，ビットベクトル同士のハミング距離の計算は，ビットベクトル間の排他的論理和を計算し，1 が立っているビットをカウントするだけで済むため，元々の素性ベクトル間の類似度計算よりも，計算が容易である [2]．

LSH ではコサイン類似度がしきい値以上のパタン対を近似的に得る事ができる．実際に利用する際は，類似度のしきい値を実際に設定したい値よりも下げておくことにより，類似パターン対を取りこぼさないようにしておく．

[2] 1 が立っているビットをカウントするという処理は，Intel SSE 4.2 では popcnt という専用の命令で高速に実行できる．

3　分散並列処理

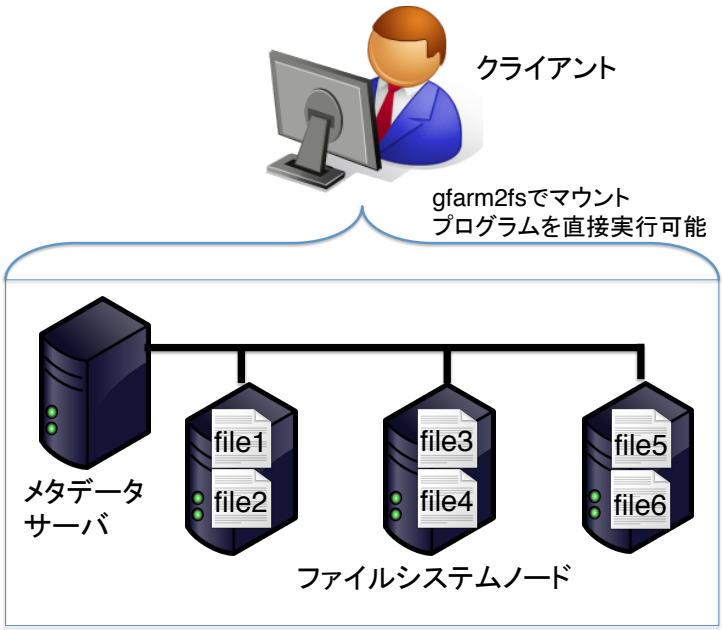
本論文では，数十億文の Web コーパスという，非常に大規模なデータを処理する．このように大規模なデータを処理する場合，データを単一のノードに配置し，そのノードだけで処理を行うことは，ハードウェアの容量の面から見ても，計算時間の面から見ても，非現実的である．このため，複数のノードにデータを分散して配置し，これを並列で処理する必要がある．本論文では，データの分散配置と並列処理を高速かつ平易に行なうために，分散ファイルシステムを利用した．分散ファイルシステムとは，複数のノードに分散して配置してあるファイルについて，ネットワークを介し，複数のマシンで共有することを可能とするファイルシステムである．本研究では，分散ファイルシステムである Gfarm[28, 29] と Hadoop[3] を利用している．本節ではこれらについて説明する．

3.1　Gfarm

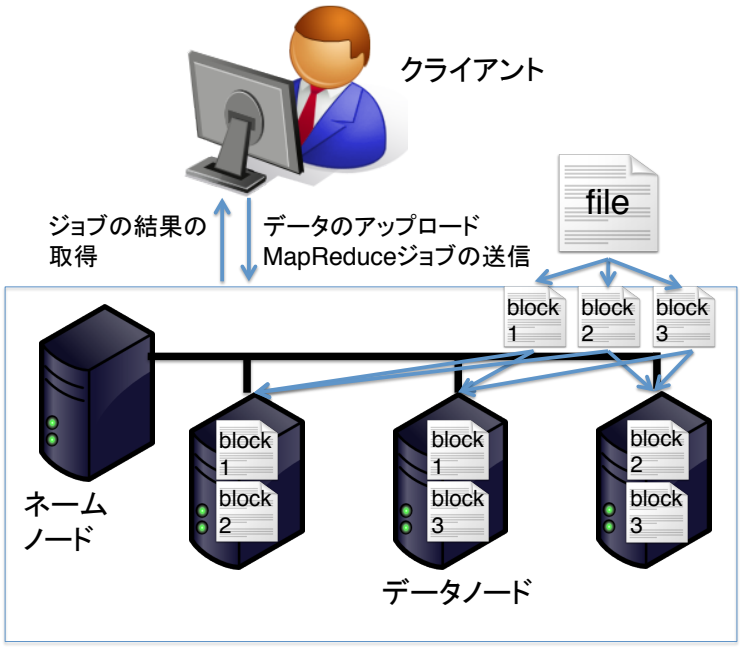
Gfarm は建部らによって開発が行なわれている，ネットワーク共有ファイルシステムである [4][28, 29]．Gfarm ファイルシステムの構成を図 1(a) に示す．図 1(a) のように，Gfarm ファイルシステムは複数のファイルシステムノードとメタデータサーバからなる．メタデータサーバは，ファイル情報，ディレクトリ情報やファイルを格納しているノード番号などのメタデータを管理する．Gfarm はこのメタデータをファイルの格納から分離して独立に管理し，さらに，メタデータをなるべくメモリに保持することで，クライアントからの要求に対し高速な応答を目指す．メタデータサーバはまた，ファイルシステムノードが利用可能かどうかやファイルシステムノードの CPU 負荷，ディスクの利用容量などファイルシステムノードの状態を管理する．メタデータの分散管理，冗長管理については実装されていないが，メタデータをバックエンドのデータベースに保持することで，突然の障害に備えている．

ファイルシステムノードは実際にファイルを格納しているノードである．Gfarm

[3] http://hadoop.apache.org
[4] http://datafarm.apgrid.org/index.ja.html



(a) Gfarm ファイルシステムの構成



(b) Hadoop 分散ファイルシステムの構成

図 1: 分散ファイルシステムの構成

のファイルシステムは，専用領域の作成を要求せず，ファイルシステムノード上の，ローカルファイルシステムの指定されたディレクトリ以下を束ねることにより，分散ファイルシステムを構成する．このような構成になっているため，ファイルシステムノードを追加すると，ファイルシステム全体の容量がその分増加することになる，すなわち，容量面でのスケールアウトを実現している，

クライアントは，gfarm2fs コマンドにより，Gfarm ファイルシステムをマウントすることで，ローカルファイルシステムのように Gfarm ファイルシステム上へのアクセスが可能となる．つまり，クライアントの視点からは，Gfarm ファイルシステムは一つの巨大なファイルシステムとして見ることができる．実際にクライアントが Gfarm ファイルシステム上のファイルにアクセスする場合には，まず，メタデータサーバにファイルの位置を問い合わせ，その後，実際にファイルを格納しているノードと直接通信し，データのやりとりを行なう．このように，データのやりとりにメタデータサーバが介在しないため，ファイルを格納しているノード上で読み書きを行なうことで，プログラムを高速に実行することができる．従って，プログラムの実行位置まで含めたタスクファイルを作成し，タスクスケジューラに投入することで，ネットワーク転送を最小限に抑えた，高速な分散並列処理を行なうことができる．

3.2 Hadoop

Hadoop は，Apache により開発されている，大規模データの分散処理を行うソフトウェアである．Hadoop 分散ファイルシステムの構成を図 1(b) に示す．Hadoop 分散ファイルシステムは図 1(b) のようにネームノードとデータノードから構成される．ネームノードは各データノードに分散して配置してあるデータのメタデータを持つ．言い換えれば，Hadoop 分散ファイルシステム上におけるデータのリスト，データの格納位置は，ネームノードが管理している．

データノードは実際に各データを格納しているノードである．図 1(b) のように，Hadoop 分散ファイルシステムは，一つの非常に大きなファイルを複数のブロックに分割し，このブロックを各ノードに分散して配置する．Hadoop 分散ファイルシステムでは，データを大きなサイズのブロック（64MB の倍数で設定）とす

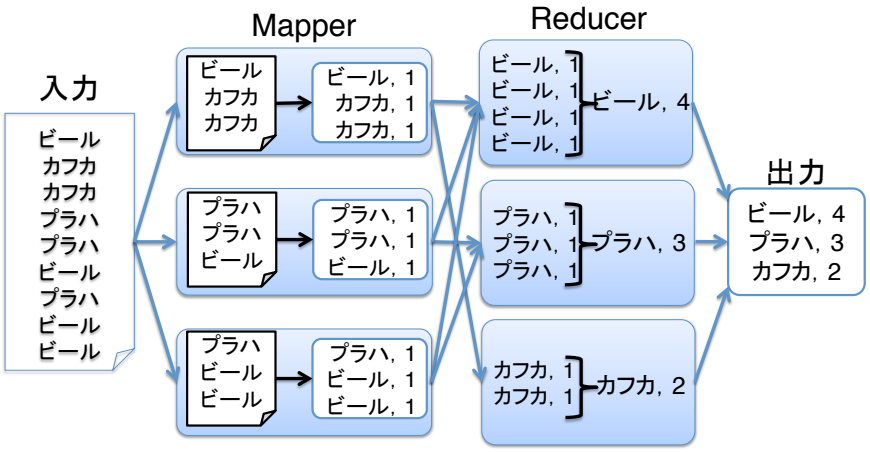


図 2: MapReduce による単語頻度のカウント

ることで，メタデータの容量と複雑さの増加を防ぎ，処理におけるスケールアウトを可能としている．また，図 1(b) に記したように，各ブロックについて，その複製を複数のノードが所持している．このため，どこかのノードに破損や障害が生じても，データが失われることはないという，信頼性，可用性を実現している．

Hadoop は Gfarm のように，マウントによって分散ファイルシステム上のファイルを扱うことができない．このため，クライアントは Hadoop ファイルシステム上のファイルに直接プログラムを実行することはできない．代わりに，図 1(b) のように，クライアントは MapReduce ジョブを Hadoop の MapReduce エンジンに送信することで，ファイルの処理を行う．すなわち，Hadoop は，MapReduce として記述される処理について，Hadoop 分散ファイルシステム上のファイルに分散処理を行う．ここで，ファイルはブロック単位で各ノードに分散しているため，入力ファイルの Hadoop ファイルシステム上への配置，MapReduce 処理，処理結果の取得には，ネットワーク越しの通信が必要である．

3.3 MapReduce

MapReduce は大規模データに対する，分散並列処理の枠組みである [30]．概要としては，まず，入力ファイルを複数に分割し，Mapper に渡す．Mapper は，

入力から，Key と Value の組を作成し（Map），結果を Reducer に引き渡す．なお，Reducer に引き渡す際，同一の Key が同一の Reduucer に渡るようにする．Reducer は同一の Key に対して，Value の集約を行い（Reduce），最終的な結果を出力する．

例として，MapReduce による単語頻度のカウントについて，図 2 に示す．この例では，一行毎に単語が書かれている入力ファイルを 3 つに分割し，Mapper に渡している．Mapper は入力を読み込み，単語とその出現を示す Key と Value，すなわち，単語の出現毎に「単語，1」という形の Key と Value を作成し，Reducer に引き渡す．例えば図 2 の一番上の Mapper では,「ビール」「カフカ」「カフカ」という単語が出現しているので,「ビール，1」「カフカ，1」「カフカ，1」を出力として，Reducer に引き渡す．Reducer は，Key 毎に Value の値の合計値を出力する，すなわち，Mapper から渡された単語の頻度を出力する．最終的に，各 Reducer の出力を集約することで，入力ファイル中の，単語頻度を獲得することができる．

自然言語処理では文書中に出現する単語や単語連続の頻度カウントなど，文書における言語現象の統計処理が必須である．実際，本論文でも，大規模コーパス中の名詞や関係パタンの頻度のカウントを必要とする．単一ノードでの処理を行う場合，入力が大規模データであると，単純な単語の頻度カウントでさえ，低頻度のものを逐次メモリから削除するというような，近似的な工夫を用いなければならない [31]．また，入力を最後まで読み込むだけでも途方もない時間がかかる．Hadoop では，クライアントは Mapper で入力からどのような Key と Value を作成するか，Reducer で Value に対しどのような処理を行うかの設計のみを行う．言い換えれば，入力ファイルを分割して Mapper に渡す，Mapper の出力を Key 毎にまとめて Reducer へ引き渡すなどが自動で行われるため，分散並列での統計処理に重宝する [5]．

[5]厳密には，ネットワーク負荷を最小限にするようなファイルサイズの設計や，Mapper，Reducer の個数などを調整しないと計算時間的に最良のパフォーマンスは得られない．

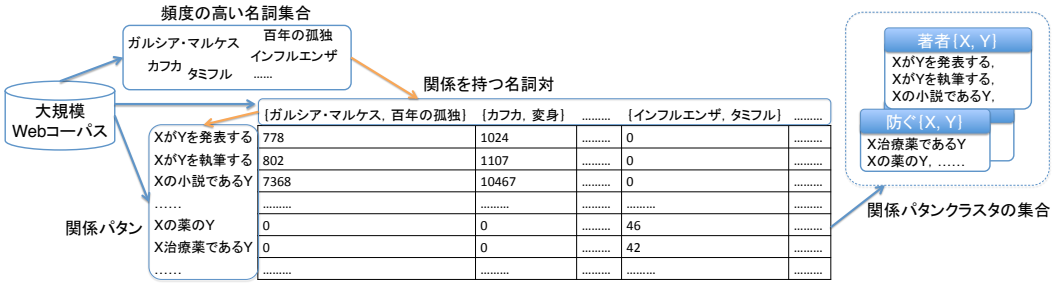


図 3: 大規模 Web コーパスからの関係パタン知識の構築

4　関係パタンの知識の構築

1 節でも触れたように，本論文では，あらゆる関係インスタンスを収集できるような関係パタンの知識を構築することを目的とする．この節では，そのような関係パタンの知識を，大規模な Web コーパスから構築する手法を提案する．具体的には，関係を持つであろう名詞対とその間の表現を抽出し，抽出した表現をクラスタリングすることによってまとめあげ，関係パタンの知識とする．

手法の概要を図 3 に示す．最初に，コーパス中で頻度の高い名詞を抽出する．次に，この名詞集合内の各名詞でペアをつくった際に，何らかの関係インスタンスとなると考えられるペアを，統計情報を元に抽出する．次に，先ほど抽出した関係を持つであろう名詞対について，コーパス中でその名詞対を結ぶ表現を関係パタンとして抽出する．最後に，名詞対と関係パタンの共起情報を元にクラスタリングを行い，同一の意味の関係パタンをまとめあげ，関係パタンの知識とする．

4.1　名詞の抽出

関係パタンを抽出する際に，その根拠となる名詞対の頻度が高くなければ，様々な種類の関係パタンを得ることはできない．コーパス中に一定以上の頻度で出現している名詞対を得るため，また，適切な名詞による名詞対を獲得することで，クラスタリングの際に素性が疎となりすぎることを防ぐために，まず，コーパス中から名詞の抽出を行う．

ところで，名詞には,「内閣総理大臣」や「永久凍土」のように，複数の連続す

る名詞で一つの名詞のような振る舞いをするものがある．しかしながら，一概に連続する名詞はすべてひとくくりとして扱えるわけではない．例えば「現状彼は困っている」のような文について，連続する名詞をひとくくりにすると,「現状彼」という，名詞として適切でないまとまりを得てしまう．

また,「道の駅」や「団塊の世代」のように,「名詞の名詞」（以下，A の B と表記する）という形で一つの名詞のように扱われるものもある．これについても，一概に A の B の形で表記されているものを抽出すると,「次の」や「たくさんの」のように，修飾するために広く用いられているものと結びつけて抽出してしまう．このため，名詞連続や A の B の形で表記されているもののうち，一つの名詞として扱うべきものをあらかじめ獲得しておく必要がある [6]．

名詞連続や A の B のように，複数の単語がまとまって一つの単語のように扱われるものは，そのまとまり内での単語同士の結びつきが，個々にバラバラな単語同士よりも強い．このため，単語間の結びつきの強いまとまりは，一つの単語として扱うことができると考えられる．本論文では，単語間の結びつきの強さを測る指標として，式 (3) で表される，自己相互情報量（PMI）のような値をスコアとして用いる．

$$score(w_i, w_j) = \begin{cases} 0 & \text{if } seq(w_i, w_j) \leq \delta \\ \frac{seq(w_i, w_j)}{freq(w_i) * freq(w_j)} * discount(w_i, w_j) & \text{else} \end{cases} \tag{3}$$

$$discount(w_i, w_j) = \frac{seq(w_i, w_j)}{seq(w_i, w_j) + 1} * \frac{\min(freq(w_i), freq(w_j))}{\min(freq(w_i), freq(w_j)) + 1} \tag{4}$$

ここで，w_i や w_j は単語であり，今回は名詞，あるいは「A の」である．$seq(w_i, w_j)$ は二つの単語 w_i と w_j が $w_i w_j$ という連続した形でコーパス中に出現した頻度であり，$freq(w_i)$ は単語 w_i の出現頻度である．また，δ は定数であり，単語連続の出現頻度のしきい値である．

δ で単語連続の出現頻度にしきい値を用いてはいるが，PMI と同様，単語 w_i や w_j の出現頻度が少ない場合に，式 (3) の $score(w_i w_j)$ が大きくなりすぎてしまう問題がある．これを防ぐために，式 (4) のような discount factor が提案されてい

[6]複数の単語で一つの名詞のように扱うものには，他にも,「排他的な関係」や「白い恋人」のように，形容詞などと結びつくものがあるが，本論文では対象を上述の二つに絞る．

る [32]．これは，単語 w_i と w_j が十分な数出現していないときや $seq(w_i, w_j)$ が小さいときに，$score(w_i, w_j)$ の値を抑える働きがある．上記の式 (3) から求められる $score(w_i, w_j)$ がしきい値を超えている場合に，$w_i w_j$ を一つのまとまり，すなわち，一つの単語として扱うこととする．

ある程度長い名詞連続を獲得するため，上記の処理を，コーパス中の名詞について，複数回（2-4 回）行う．つまり，まず二つの名詞連続からなるまとまりを獲得し，これを一つの名詞として考えて再び名詞連続を獲得し，またこれを一つの名詞として考えて名詞連続を獲得し，という処理を繰り返す．A の B の形式については，A と B にも名詞連続が入る可能性があるが，処理開始時点では一つの単語として扱う名詞連続が明らかでないため，名詞連続をまず獲得し，最後に A の B を獲得する．

最後に，よく使われる名詞を獲得するため，一つの名詞として扱うこととした名詞連続と A の B を含めて，コーパス中での名詞の出現頻度をカウントする．コーパス中での出現頻度が上位 N 個の名詞を対象の名詞として獲得する．本論文では N =100 万とした．

4.2 名詞対の抽出

有用な関係パタンの獲得が可能であるよう，4.1 節で抽出した名詞を元に，関係インスタンスと考えられる名詞対を抽出する．関係インスタンスである名詞対は，互いに相関があると考えられるため，共起頻度や PMI のような統計的指標を用いて名詞間の相関を測定し，抽出を行う．

「1967 年，ガルシア・マルケスは百年の孤独を発表した」という文における {ガルシア・マルケス，百年の孤独} や「インフルエンザの薬であるタミフルの供給が足りなくなっている」という文の {タミフル，インフルエンザ} のように，関係インスタンスである名詞対は同一の文に出現することが多いと考えられる．そこで，関係インスタンスであるかどうかを識別するために，最も単純な指標として，同一の文内に出現することを共起していると考えたときの，共起頻度を用いることが考えられる．すなわち，この共起頻度が高い名詞対を，関係インスタンスである可能性が高いとして抽出する．

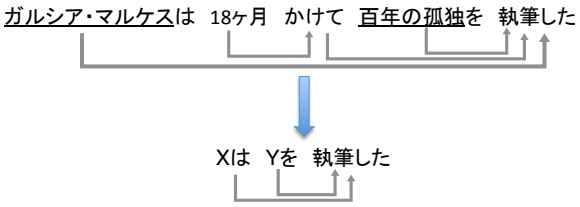


図 4: 係り受け関係に基づいた関係パタンの抽出

しかしながら，共起頻度のみに基づいた場合，{私，いつか} や {自分，今日} のように，名詞単体での出現頻度は高いが，関係インスタンスではないペアを抽出してしまう問題がある．そこで，4.1 節で名詞連続を獲得するときに用いたように，相互の結びつきの強さを測る式 (3) を適用する．ただし，4.1 節での式 (3) では，名詞連続や A の B を獲得するために順序も考慮していたが，今回は名詞間の相関を測るのが目的であるため，順序に関する制約がない．したがって，ここでは，式 (3) における $seq(w_i, w_j)$ を名詞 w_i と w_j の共起頻度で置き換えたものを $score(w_i, w_j)$ として利用する．すなわち，名詞 w_i と w_j の共起頻度を $co(w_i, w_j)$ とすると，$co(w_i, w_j) = co(w_j, w_i)$ であるため，$score(w_i, w_j) = score(w_j, w_i)$ である．この $score(w_i, w_j)$ に基づき，上位 M 個の名詞対を抽出する．本論文では，M =100 万とした．なお，先述のように，相関の高い名詞対を抽出することが目的であり，順序は考慮しないため，$\{w_i, w_j\}$ と $\{w_j, w_i\}$ は同一のものとして扱う．

4.3 関係パタンの抽出

4.2 節で抽出した，関係インスタンスと考えられる名詞対の間を，文内で結ぶ表現を関係パタンとして抽出する．日本語は,「ガルシア・マルケスは 18ヶ月かけて百年の孤独を執筆した」と「ガルシア・マルケスは百年の孤独を 18ヶ月かけて執筆した」のように，単語の順序についての制約が緩い．そのため，文字列の並び順序をそのままパタンとした場合，これらの文から「X は 18ヶ月かけて Y を執筆した」と「X は Y を 18ヶ月かけて執筆した」という二種類のパタンが抽出されてしまう．このように，文字列の並び順序を直接パタンとして扱うと，その数は

非常に膨大なものになってしまい，パタンを用いてインスタンスを抽出する際に，マッチングが行えないパタンがほとんどであるというような不便が生じる．これを解決するため，本論文では，係り受け関係を用いて関係パタンを表すこととする．具体的には，対象の名詞対の間の係り受けパスをパタンとして抽出する．なお，係り受け解析には既存のツールである，cabocha を用いることとする．

係り受け関係に基づいた関係パタンの例を図 4 に示す．図 4 では,「ガルシア・マルケスは 18ヶ月かけて百年の孤独を執筆した」という文について，各文節同士の係り受け関係を文節間に記した矢印で表している．この文に対象の名詞対が含まれる場合，その名詞を含んでいる文節間の係り受けパスを関係パタンとして抽出する．また，関係パタンとして抽出する際に，抽出の根拠とした名詞対は，それぞれ X，Y のように変数化する．例えば，図 4 の文が与えられたとき, {ガルシア・マルケス，百年の孤独} が対象の名詞対であったとすると，これらの名詞を含む文節はどちらも「執筆した」という文節にかかる．このため，係り受けパスを抽出し，名詞部分を変数化することで，図 4 の下部に記したように,「X は→執筆した_Y を→執筆した」というパタンを得られる．なお，図 4 では簡便のために省いているが，実際には，単語は基本形に変換しており，また，各単語に品詞情報を付与して抽出し，関係パタンとしている．

従来自然言語処理が係り受け解析など構造解析の対象としていた新聞と比べ，Web コーパス上には口語表現や，記号の使用などが多いため，係り受け解析をしばしば誤ってしまう．これにより，名詞対を結ぶ係り受け表現を列挙すると，意味をなさない表現や，誤った係り受けパスを抽出してしまうことがある．この誤った係り受けパスを除去するため，頻度がしきい値 ϕ 回以上のパタンのみを抽出する．この頻度は，対象の名詞対との共起頻度の合計とする．例えば,「X は→執筆した_Y を→発表した」というパタンについて，X，Y のそれぞれに {ガルシア・マルケス，百年の孤独} が項となる回数が 778 回であり, {カフカ，変身} が項となる回数が 1,024 回であって，それ以外の名詞対は項にならなかった場合，この関係パタンの頻度は 1,802 となる．本論文では，ϕ を 50 とした．

ここで，関係を表現していると考えられるパタンを抽出するのであれば，名詞対の抽出に用いたように，頻度ではなく，式 (3) のようなスコアに基づくべきで

はないかという疑問があるかもしれない．しかしながら，式 (3) に基づいて計算したスコアを適用すると,「X の→ヒューゴー賞受賞作である→ Y」のように，項となる名詞対が少ないものを獲得してしまう．このため，大量の関係インスタンスを獲得できるような関係パタンの知識が構築できない恐れがある．より幅広く適用できる関係パタンを得るため，本論文では，関係パタンの頻度を計算し，頻度の高いものをクラスタリング対象とする．

4.4 クラスタリング

関係パタンの抽出を行っただけでは，各パタンがどのような種類の関係を表すかが明らかになっていない．例えば,「X は Y を発表した」と「X は Y を執筆した」の二つのパタンが共に「X は Y の著者である」という関係を表すと判断することができない．これを解消するため，抽出してきた関係パタンをクラスタリングし，関係の種類毎にパタンをまとめあげる．

4.3 節では対象の名詞対を結ぶ係り受けパスを列挙し，頻度上位のものを取得しているため,「X の Y」や「X による Y」のような，幅広く使われるパタンも獲得される．このように述語を含まないパタンは，Open IE における先行研究では扱っていない [9, 10]．しかし，あらゆる関係インスタンスの抽出を目的とする場合，ロングテールを拾い上げるために，このようなパタンも有用であると考えられる．

これら述語を含まないパタンは「ガルシア・マルケスによる百年の弧度」（**著者**｛ガルシア・マルケス，百年の孤独｝）や「飲酒運転による事故」（**因果**｛飲酒運転，事故｝）のように，様々な種類の関係を表すため，直接クラスタリングに適用することは難しい．ここで,「作家による小説」であれば著者関係,「行為による現象」であれば因果関係というように，名詞の種類（クラス）に着目することで，どのような関係を表すパタンなのか識別できると考えられる．

従って，本手法では，まず対象としている名詞についてクラスタリングを行い，名詞のクラスを獲得する．次に，この名詞クラスを利用してパタンの語義を識別し，語義毎に分解してクラスタリングを行う．

4.4.1 名詞のクラスタリング

4.1 節で抽出した N 個の名詞についてクラスタリングを行い，名詞のクラスを獲得する．ここで，対象としている名詞は非常に大量に存在するため，クラスタリングに用いる素性を高次元にしてしまうと，計算時間が膨大になってしまう．本論文では，word2vec というツール[7]を用いることで，次元数の小さな，高品質のベクトルを獲得し，これを素性として用いる．

word2vec は，コーパスから，固定次元長の単語の特徴ベクトルを学習するツールである．学習は，ニューラルネットワーク言語モデルから派生した，Skip-gram モデルというモデルについて行い，質の高い特徴ベクトルを出力する [33]．Skip-gram モデルとは，ある単語について，文中でその単語の周辺に出現する単語のベクトルを予測できるように，単語ベクトルを学習するモデルである．具体的には，単語列 w_1，w_2，$...w_T$ について，式 (5) の値を最大化する．

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t) \tag{5}$$

ここで，c は単語 t について，周辺の単語としてどの程度までを対象にするか，という値である．すなわち，式 (5) は，各単語 w_t について，範囲 c にある単語の出現確率を最大化する．$p(w_{t+j}|w_t)$ については，ソフトマックス関数を用い，次の式 (6) のように定義している．

$$p(w_{t+j}|w_t) = \frac{\exp(v_{w_{t+j}}) \cdot v_{w_t}}{\sum_{w'=1}^{W} \exp(v_{w'} \cdot v_{w_t})} \tag{6}$$

ここで，v_w は単語 w のベクトルであり，W は語彙数である．学習は，すべての文，およびその中の単語について，式 (6) に対応する勾配を求め，確率的勾配降下法と誤差逆伝播法で単語ベクトル v_{w_t} と単語予測ベクトル $v_{w_{t+j}}$ を更新する．

本論文では，word2vec の学習データとして，Web コーパス中の各文から内容語，具体的には名詞と，動詞，形容詞，名詞+助動詞の「だ」あるいは「です」（犬は動物だの「動物だ」の部分）など述語のみを抽出したデータを用いる．ここで，学習時間短縮のため，内容語の少ない文は除去している．また，word2vec

[7] https://code.google.com/p/word2vec

では学習の際に，高頻度語の subsampling を行うことや，疑似負例のサンプリングを行うことで，学習の速度と精度を向上させているが，ここでは割愛する．

名詞のクラスタリングには，クラスタリングアルゴリズムとして一般的な k-means アルゴリズムを用いる [34]．出力のクラスタ数を k としたとき，k-means アルゴリズムによる名詞のクラスタリング手順は以下のとおりである．

1. 各名詞をランダムに k 個のうちどれか一つのクラスタに割り当てる．
2. 各クラスタに属する名詞のベクトルの平均値を計算し，そのクラスタのセントロイドとする．
3. 各名詞について，距離が最小となるセントロイドのクラスタに割り当てる．
4. 前の反復からクラスタに変化がない（収束）か，反復回数の最大値に至った場合，結果として，各名詞の現在所属するクラスタを出力する．そうでなければ，上記の 2 と 3 を繰り返す．

本論文では，距離の指標として，入力を L2 ノルムで正規化したユークリッド距離を用いる．

k-means アルゴリズムの出力は，収束時間や出力の質が，ランダムなクラスタの初期化に大きく依存する．計算時間と出力クラスタの質の両方の面においてこの問題を解決するために，初期値を工夫する手法が提案されている [35]．これは，初期の k 個のセントロイドは互いになるべく離れていた方が良いというアイデアに基づいている．初期値決定の概要としては，まず始めに，入力データのうちランダムに 1 つを選びセントロイドとする．次に，入力の各データについて，最も近いセントロイドを計算し，その距離の二乗に比例した確率に従い，ランダムにセントロイドを選択する，という手順を繰り返すものである．本論文でも初期値の選択にこの手法を適用した．

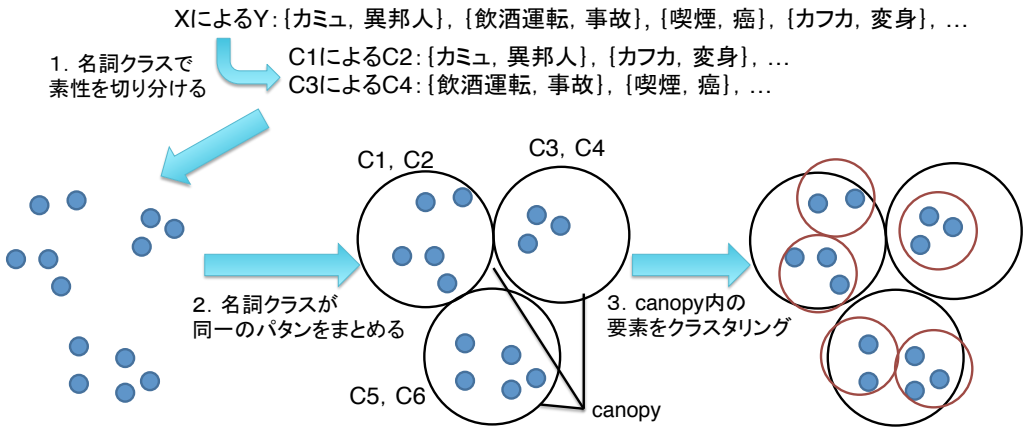


図 5: 名詞クラスを用いた関係パタンのクラスタリング

4.4.2 関係パタンのクラスタリング

パタンのクラスタリングは，意味的に似ている語句はその出現文脈も似ているという，分布仮説に基づいて行う [36]．すなわち，パタンの共起する名詞対をパタンの文脈と考え，共起する名詞対とその頻度を素性として，類似度の計算を行う．これにより，例えば，「X は Y を発表した」と「X は Y を執筆した」は共に {ガルシア・マルケス，百年の孤独} や {カフカ，変身} など作家と小説の名詞対と共起することから，類似度が高くなり，同一の関係を表すパタンとしてまとめることができる．

しかしながら，既に記したように，パタンには，「X による Y」のように，多義性のあるものも存在する．また，類似度の計算について，単純に全パタン間の類似度を計算すると，その計算量は $O(n^2)$ である．これは，今回のように n が非常に大きな値である場合には，実時間での計算が到底不可能な値である．そこで，本手法では，名詞クラスを用いてあらかじめ同一の関係であると考えられる名詞を近似的にまとめあげ，その後，改めてクラスタリングを行うという手法をとる事で，パタンの多義性と，計算時間の問題を解決する．

関係パタンのクラスタリングの概要を図 5 に示す．本手法では，まず，各パタンについて，共起する名詞対のクラスで素性を切り分ける．すなわち，図 5 のよ

うな「X による Y」について,{カミュ，異邦人} や {カフカ，変身} のように作家と小説の名詞を含むクラスである $C1, C2$ と,{飲酒運転，事故} と {喫煙，癌} のように，行為と現象の名詞を含むクラスである $C3, C4$ の名詞対に分ける．これにより,「X による Y」というパタンを「X は Y の著者である」という関係と,「X は Y の原因である」という関係に分ける事ができる．次に，この素性を切り分けたパタンを入力とし，X と Y に同一のクラスを持つパタン毎にまとめる．図 5 では，入力をそれぞれ，$C1, C2$，$C3, C4$，$C5, C6$ のクラス対を持つパタンに分けられる．この識別した結果の各集合を canopy と呼ぶ．最後に canopy 内の要素を正確な類似度計算でクラスタリングし，最終的な出力を得る．この canopy を作成し，canopy 内でのクラスタリングを行うという手法は，canopy の数を c とすると，計算量を $O(n^2)$ から $O(n^2/c)$ に削減できることが知られている [37].

canopy 内でのクラスタリングについては，階層的クラスタリング手法である，群平均法を用いる．類似度の指標には式 (7) のコサイン類似度を用いる．

$$\cos(p, q) = \frac{\sum_{i=1} p_i \cdot q_i}{\sqrt{\sum_i p_i^2} \cdot \sqrt{\sum_i q_i^2}} \tag{7}$$

ここで，p，q は共に同一の canopy 内に含まれるパタンであるとする．群平均法の際の類似度のしきい値としては α を用いてクラスタリングを行う．本論文では，$\alpha = 0.3$ とする．

5　評価実験

5.1　実験設定

本節では，構築した関係パタンの知識の質を，実験をとおして検証する．実験では，まず名詞クラスタリング結果の質を評価し，その後，この名詞クラスタを利用して獲得した関係パタンの知識の質を評価する．名詞クラスタの評価については k-means アルゴリズムの k の個数を変化させたときの評価を行う．パタンについては，名詞クラスタを利用し，canopy クラスタリングを行った結果と，LSH による高速な近似近傍探索手法を利用した類似度計算によりクラスタリングを行った結果とを比べる．すなわち，名詞クラスタを利用する事で「X の Y」のような複数の関係を表す表現の意味の分解を行い，さらに類似度計算対象を canopy 内に絞る事で高速化した手法と，意味の分解を行わず，全パタン間の類似度を LSH を用いて計算した手法との比較を行う．LSH のビット数は 1024 とし，コサイン類似度 0.1 以上のパタンペアを抽出する．パタンの評価では，クラスタリングの質の評価に加えて，パタンの類似度計算時間の比較として，LSH におけるビットベクトルへの変換，近傍点の探索，しきい値以上のパタンペアの正確な類似度計算時間の合計と canopy クラスタリングにおける類似度計算時間との比較を行う．

名詞クラスタリングの評価については，Wikipedia の infobox から情報を抽出したデータセットである，DBpedia[8] と関根の拡張階層固有表現を元に [9]，機械的な手法でいくつかクラスを作成し，これとの比較を行う．パタンのクラスタリングの質については，DBpedia から機械的に抽出した関係インスタンスや人手で正否をつけた関係インスタンス集合を用いて，評価を行う．具体的には，DBpedia から単純なルールで抽出できる関係インスタンスとして,「人名が作品を書いた」という著作関係,「会社（人物）が物を製作した」という製造品関係,「場所（建物）が場所に存在する」という所在地関係のインスタンスを抽出した．また,「ある物（現象）がある現象を引き起こす」という因果関係,「ある物（現象）がある現象を防ぐ」という予防関係について，対象の名詞対の中からランダムにサンプ

[8] http://ja.DBpedia.org/
[9] https://sites.google.com/site/extendednamedentityhierarchy/

リングを行い，人手で正否を判定した．この正解データを用い，本研究で獲得した関係パタンのクラスタがどの程度関係インスタンスを獲得できるか評価する．

ところで，今回の手法では，相関の強い名詞対100万対のみがインスタンス候補となっているため，この候補からもれた名詞対は抽出できない．そこで，DBpediaから抽出したインスタンスで，かつ100万の名詞対に含まれるものに限定して評価したいが，これを行うと，正解インスタンス数が非常に少なくなってしまう．これを解消するため，DBpediaから抽出した著作関係，製造品関係，所在地関係について，システムが出力したクラスタのうち，DBpediaから得た正解の関係インスタンスを含んでいた複数のクラスタから，ランダムに名詞対のサンプリングを行い，人手で正否を判定したうえで，正解データに加えた．最終的に，著作関係，製造品関係，所在地関係，因果関係，予防関係について，それぞれ610個，371個，840個，169個，140個の正解インスタンスを得た．

実験には，日本語Web文書約60億文をコーパスとして用いた．この文書は日本語係り受け解析器であるCaboCha[10]によって係り受け構造を解析済みである．なお，実験対象のパタンである，対象の名詞対との共起頻度の合計50以上のものは約50万個（494,799個）であった．

5.2 実験結果

5.2.1 名詞クラスタリング

表 1: 名詞クラスタリングの結果と作成のための計算時間

k	適合率 (%)	再現率 (%)	F1	計算時間 (h)
500	16.3	8.2	10.9	16
1000	21.2	5.1	8.2	55
1500	24.9	4.1	7.0	74

名詞クラスタリングについて，機械的な手法で作成した10種類のクラス，49,746個の名詞からなるデータセットを正解データとして用い，評価を行った．クラス

[10]https://code.google.com/p/cabocha

表 2: 対象の名詞に対し，類似度の高い名詞

対象の名詞	類似度の高い名詞
プリウス	新型プリウス，ハイブリッド車，ハイブリッド専用車， シビックハイブリッド，インサイト，トヨタのプリウス， レクサス，７車種，ハイブリッドカー，カムリ
風邪	鼻風邪，夏風邪，カゼ，風邪気味，喉風邪， 風邪ひき，風邪引き，風邪の症状，喉の痛み，咳
芥川龍之介	森鴎外，芥川竜之介，夏目漱石，川端康成，志賀直哉， 谷崎潤一郎，太宰治，文豪，永井荷風，菊池寛
川上弘美	センセイの鞄，新潮文庫，村上春樹，高橋源一郎，山田詠美， 山本文緒，重松清，宮部みゆき，江國香織，川上弘美さん

タリング結果の各クラスタについて，正解データの各クラスの要素をどの程度含むか計算し，各クラスについて，最も多く要素を含むクラスタをそのクラスであると考えて，適合率と再現率を計算した．結果を表 1 に示す．

表 1 のように，k-means における k の数を増加させていけば，適合率が上がり，再現率は下がる．これは，k の値を増やす，すなわち，クラスタ数を増やす事で，各クラスタは少数の，非常に近いベクトルを持つ名詞の集合になるためであると考えられる．しかしながら，表 1 の F1 値から分かるように，適合率の上昇よりも再現率の減少の方が深刻であり，作成にかかる計算時間の面から見ても，名詞クラスタリングにおいては，k の値は 500 が妥当であると考えられる．

表 1 の結果について，どの k についても，適合率や再現率の値がそれほど高くないが，これは，正解データを別のコーパスから機械的につくっているため，表記揺れのマッチングが取れていない事に問題があると考えられる．さらに，正解データの抽出元は Wikipedia であるため，施設名において「ヤマザキマザック美術館」や病名において「ウォーターハウス・フリードリヒセン症候群」のような，Web 文書中での頻度上位 100 万に入らない名詞が正解に多く含まれている．このような，正解データにおける，クラスタリング対象となっていない名詞の多さが，再現率の上がらない一因として考えられる．

いくつかの名詞について，クラスタリングに用いた単語特徴ベクトルで類似度を計算し，類似度の高い順に単語を 10 個抽出した結果を表 2 に示す．表 2 に示したように，word2vec により学習した単語ベクトルを用いて類似度を計算すると，およそ同義と言って良い単語が類似度の高い単語として抽出される．例えば，車種である「プリウス」に対しては,「インサイト」や「ハイブリッド車」,「シビックハイブリッド」など，車種を表す語が，類似度の高い語として抽出される．しかしながら，作家である「川上弘美」について，その作品名である「センセイの鞄」や出版社である「新潮文庫」が類似度の高い語として出現してしまっている．これは，word2vec が，周辺の単語のベクトルを予測できるように単語ベクトルを学習する，というモデルであるため，相関の高い語も似た単語ベクトルを獲得してしまったのだと考えられれる．このように，上位下位ではなく，トピックのようなレベルで同じ単語が似た単語ベクトルを獲得している事も，適合率が上がらない原因の一つであると考えられる．すなわち，正解データは関根の拡張固有表現と wikipedia を元にしており，いわば上位下位関係に基づいて同義の単語集合を形成しているが，word2vec によって学習した単語ベクトルでは，同一の文に出現しやすい単語など，相関の高い単語の類似度も高くなり，正解データとの差異が生じたのだと考えられる．

5.2.2 関係パタンのクラスタリング

表 3: 関係パタンのクラスタリング結果の比較

関係	LSH		Canopy クラスタリング	
	適合率 (%)	再現率 (%)	適合率 (%)	再現率 (%)
著作	16.2	14.6	85.2	11.3
製造品	51.1	48.8	58.3	20.8
所在地	66.0	32.9	47.6	27.6
因果関係	72.7	28.4	100	13.1
予防	8.2	17.1	40.8	14.3

LSH による高速な近似近傍探索を利用して，全パタン間の類似度を計算し，ク

表 4: 関係パタンのクラスタリングで生成されたクラスタ数および類似度計算時間

LSH		Canopy クラスタリング	
クラスタ数 (個)	計算時間 (h)	クラスタ数 (個)	計算時間 (h)
174,509	62.03	462,605	0.62

ラスタリングを行ったときと，k=500 の設定で作成した名詞クラスタを利用して，関係パタンから canopy を作成し，その canopy 内で名詞対を素性にクラスタリングしたときの，クラスタリング結果について表 3 に示す．クラスタリングでは明示的なラベルが付与されないため，どのクラスタがどの関係と対応づくか不明である．正解データに含まれる各関係と生成されたクラスタを対応づけるため，各クラスタと正解データとの F1 スコアを測定し，各関係について最も高いスコアを出したクラスタを対応づけた．表 3 にはさらに，5 つの関係についての適合率，再現率の合計値を記してある．

表 3 から，LSH に比べ，Canopy クラスタリングでは適合率が上昇している事が見て取れる．これは，名詞クラスタを利用する事により，関係パタンの意味がより洗練されたためであると思われる．すなわち，名詞クラスタを利用して canopy を作成する事により，複数の意味を持つ関係パタンの意味が分離できたためであると考えられる．反対に，Canpy クラスタリングでは，再現率が低下してしまっている．これは，同じ関係を表すにも関わらず，クラスタから分離されてしまった関係パタンが存在する事を示唆している．

なお，所在地関係では適合率，再現率共に LSH よりも下回っているが，実際に関係インスタンスを人手で見ると，所在地関係であるものがほとんどであった．これは，自動的手法とサンプリングによって正解データを作成しているために発生したと考えられる．

LSH による手法と Canopy クラスタリングとで，生成されたクラスタ数および類似度計算時間について，表 4 に示す．なお，ここでは，クラスタ内に関係パタンを 10 個以上含むクラスタのみに絞って個数を数えている．表 4 から，Canopy クラスタリングでは，類似度計算時間が大幅に減少している事が分かる．今回は約 50 万パタンを対象にクラスタリングを行ったが，この結果から，100 万パタン

表 5: Canopy クラスタリングで獲得した関係パタンの例

関係	パタンの例
著作	X の作者である Y，X で有名な Y，X の生みの親 Y
製造品	X は Y をマイナーチェンジした，X の新型 Y，X のミニバン Y
所在地	X を Y で探せます，X で探す Y の引っ越し業者， X の賃貸物件をお探しなら Y の賃貸が充実
因果関係	X の原因は Y によるものです，X の原因が Y，X の原因となる Y
予防	X を和らげる Y，X を鎮める Y，X には Y を使う

など，対象を増やしても対応できる事が期待できる．

LSH と比べると，Canopy クラスタリングでは生成されたクラスタ数が多い．これは，同じ関係を表す関係パタンが別々のクラスタになってしまっているケースが多いためであると考えられる．Canopy クラスタリングでは，異なる canopy に属するパタン，すなわち，項の名詞クラスタが異なる関係パタンは確実に分離されてしまう．このため，名詞クラスタが仔細に細分化されているなどして必要以上に多い場合，クラスタリング結果の再現率が低下してしまうことが考えられる．また，canopy からクラスタリングを行った際に，同じ関係を表す関係パタンを分離してしまっている事も考えられる．名詞クラスの作成方法を変える事や既存の言語資源を利用する事，また，クラスタリングの際の素性を密にするなどして生成されるクラスタの細分化を防ぐ事は，今後の課題である．

Canopy クラスタリングによって獲得したパタンの例を表 5 に示す．表 5 では，簡略のために，品詞や係り受け情報は除いており，また，単語を基本形になおすという処理も行っていない．この表から，クラスタリングによって，各関係を表すパタンを獲得できている事が分かる．また，例えば著作関係の「X の生みの親 Y」という関係パタンは,「メトロイドの生みの親任天堂」のように製造品関係も表すが，名詞クラスタを利用して関係パタンの意味を分離する事により，人と創作物という，著作関係の関係パタンとして扱えている．

Web 文書から関係パタンを抽出し，クラスタリングを行うと，所在地関係の関係パタンのように，限定的な関係パタンのみのクラスタを作成してしまう事もあ

る．これらの関係パタンは,{由布市，大分県} のように，所在地関係のインスタンスとよく共起するので，関係インスタンスの獲得を目的とした場合は有用であるが，意味解析への適用など，言語的な知識資源として有用であるかは疑問である．今後，これらの関係パタン内の単語の組み合わせで，どのようにして関係を表すのかを計算する枠組みを構築することは，ロングテールへの対応や意味解析への応用などの点から重要であると考えられる．

6　まとめ

本論文では，大量の Web 文書から大規模な関係パタンの知識を構築する手法を提案した．特に，分散並列での計算を実行する事により，約 60 億文という，大規模な Web 文書から，実用的な時間で知識の獲得を実現した．さらに，あらかじめ関係のありそうな名詞対を取得しておく事，名詞クラスタを用いてパタンの多義性を解消する事により，既存の Open IE の研究では着目されていなかった,「X の Y」や「X による Y」のような，述語を含まない表現の，関係パタンとしての利用を可能にした．

本研究では，クラスタリングの際に，関係パタンと共起する名詞対を素性として利用しているが，大規模 Web 文書をコーパスとしていても，この素性が疎である問題は残る．今後は，素性が疎である問題を解消するために，クラスタリングの際に学習した名詞の単語ベクトルを利用することや，関係パタン内の単語を利用する事などを考えたい．さらに，より広範な表現を扱う事ができるよう,「X は Y の発生するリスクを下げる」と「X は Y を予防する」という表現が共に同一であるということを，単語単位の組み合わせで計算できるような手法を構築したい．すなわち，今後の方針として，今回取得した関係パタンの同義関係の知識を元に，単語の構成性を扱えるようなモデルを構築していきたい．

謝辞

本研究を進めるにあたって，多くの方にご協力をいただきました．ここに，心より感謝の意を表します．

乾健太郎教授には，お忙しい中，研究活動全般にわたり，終始手厚いご指導，ご助言をいただきました．心より感謝を申し上げます．ご多忙の中，審査委員をお引受けくださった，大町真一郎教授，木下哲男教授に深く感謝致します．本研究を進めるにあたり，適切なご助言をくださいました，岡崎直観准教授，渡邉陽太郎助教，松林優一郎研究特任助教，水野淳太研究員，井之上直也研究員に深く感謝致します．また，言語現象や実験結果について，洞察を与えてくださいました，菅野美和技術補佐員，福原裕一研究員に感謝いたします．大規模 Web 文書を扱うにあたり，Hadoop や Gfarm システムの立ち上げには，山口健史研究員に多くの面でご助言，お力添えいただきました．深く感謝いたします．また，研究活動および大学生活を暖かく支えてくださいました，八巻智子秘書に感謝致します．

最後になりましたが，研究室での生活から研究に関しての議論まで，多くの面で研究活動を支えてくださった乾・岡崎研究室の皆様に心より感謝致します．

参考文献

[1] Shinzato Keiji, Shibata Tomohide, Kawahara Daisuke, and Kurohashi Sadao. Tsubaki: An open search engine infrastructure for developing information access methodology. 情報処理学会論文誌, Vol. 52, No. 12, p. 12p, dec 2011.

[2] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1608–1618, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[3] Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine reading. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, pp. 1517–1519. AAAI Press, 2006.

[4] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, pp. 1044–1049. AAAI Press, 1996.

[5] Sergey Brin. Extracting patterns and relations from the world wide web. In *Selected Papers from the International Workshop on The World Wide Web and Databases*, pp. 172–183. Springer-Verlag, 1999.

[6] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 85–94. ACM, 2000.

[7] Patrick Pantel and Marco Pennacchiotti. 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. Association for Computational Linguistics, 2006.

[8] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, pp. 3–10. AAAI Press, 2011.

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

[10] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523–534. Association for Computational Linguistics, 2012.

[11] Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. Open information extraction with tree kernels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 868–877, 2013.

[12] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 25–26. Association for Computational Linguistics, 2007.

[13] Stijn De Saeger, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, and Masaki Murata. Large scale relation acquisition using class dependent patterns. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pp. 764–769. IEEE Computer Society, 2009.

[14] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pp. 52–60. Association for Computational Linguistics, 2010.

[15] Jing Jiang and Chengxiang Zhai. A systematic exploration of the feature space for relation extraction. In Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT '07, pp. 113–120, 2007.

[16] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2004.

[17] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, Vol. 3, pp. 1083–1106, mar 2003.

[18] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 1306–1313, 2010.

[19] James R. Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pp. 172–180, 2007.

[20] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information

extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 101–110. ACM, 2010.

[21] Vishnu Vyas, Patrick Pantel, and Eric Crestan. Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 225–234, 2009.

[22] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 118–127. Association for Computational Linguistics, 2010.

[23] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1027–1037. Association for Computational Linguistics, 2012.

[24] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1135–1145. Association for Computational Linguistics, 2012.

[25] Dekang Lin and Patrick Pantel. Dirt-discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 323–328. ACM, 2001.

[26] Stanley Kok and Pedro Domingos. Extracting semantic networks from text via relational clustering. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pp. 624–639. Springer-Verlag, 2008.

[27] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, pp. 380–388. ACM, 2002.

[28] 修見建部, 哲之曽田. 広域分散ファイルシステム gfarm v2 の実装と評価 (グリッド i). 情報処理学会研究報告. [ハイパフォーマンスコンピューティング], Vol. 2007, No. 122, pp. 7–12, dec 2007.

[29] Osamu Tatebe, Kohei Hiraga, and Noriyuki Soda. Gfarm grid file system. *New Generation Comput.*, Vol. 28, No. 3, pp. 257–275, 2010.

[30] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, pp. 10–10. USENIX Association, 2004.

[31] Graham Cormode and Marios Hadjieleftheriou. Methods for finding frequent items in data streams. *The VLDB Journal*, Vol. 19, No. 1, pp. 3–20, February 2010.

[32] Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *Proceedings of Human Language Technology/North American chapter of the Association for Computational Linguistics (HLT/NAACL-04)*, pp. 321–328, 2004.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.j.c. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. 2013.

[34] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedungs of the*

fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 281–297. University of California Press, 1967.

[35] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[36] Zellig Harris. Distributional structure. *Word*, Vol. 10, No. 23, pp. 146–162, 1954.

[37] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178. ACM, 2000.

発表文献一覧

学術論文誌

- 高瀬翔，岡崎直観，乾健太郎．カテゴリ間の兄弟関係を活用した集合拡張．自然言語処理，vol. 20, No. 2，pp.273-296，2013．

国際会議論文

- Sho Takase, Naoaki Okazaki and Kentaro Inui. Set Expansion using Sibling Relations between Semantic Categories. In Proceedings of the 26th Pacific Asia Conference on Language Information and Computing, pp. 567-576, 2012.

- Sho Takase, Akiko Murakami, Miki Enoki, Naoaki Okazaki and Kentaro Inui. Detecting Chronic Critics Based on Sentiment Polarity and User 's Behavior in Social Media. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop, pp.110-116, 2013.

国内会議・研究会論文

- 高瀬翔，岡崎直観，乾健太郎．意味カテゴリの階層関係を活用した集合拡張．言語処理学会第 18 回年次大会論文集，pp.475-478，2012．

- 高瀬翔，岡崎直観，乾健太郎．名詞カテゴリからの関係知識獲得に向けて．NLP 若手の会 第 7 回シンポジウム，2012．

- 高瀬翔，村上明子，榎美紀，岡崎直観，乾健太郎．ソーシャルメディア上の発言とユーザー間の関係を利用した批判的ユーザーの抽出．言語処理学会第 19 回年次大会，pp260-263，2013．