2011年度　卒 業 論 文

意味カテゴリの階層関係を活用した集合拡張

2012年3月26日

情報知能システム総合学科
(学籍番号: A8TB2114)

高瀬翔

東北大学システム工学部

概 要

少数の意味カテゴリに属する語（インスタンス）を入力とし，それらと同一の意味カテゴリに属する未知のインスタンスを新たに獲得するタスクを集合拡張と呼ぶ．集合拡張を行う際に最大の問題となるのが，本来対象としている意味カテゴリとは異なるインスタンスを獲得してしまう意味ドリフト問題である．意味ドリフト問題を解決するために，近年，様々な研究がなされてきたが，最先端の手法によっても意味ドリフトを完全に防ぐことはできていない．

本稿では意味ドリフトを抑制するために，比較的入手がしやすいカテゴリ間の上位下位情報を用いる手法を提案する．上位カテゴリが共通の意味カテゴリ（兄弟カテゴリ）を用いてインスタンスが上位概念を保有しているかを検証するためのパターンを獲得し，インスタンスを獲得する際にこのパターンによってフィルタリングを行うことにより，誤ったインスタンスの獲得を防ぐ．

実験によって最新の手法との比較を行い，提案手法は意味ドリフト問題の抑制に効果があることを示す．さらに結果から得られた意味ドリフト問題に対処するための新たな知己についても述べる．

目 次

第1章　序論

意味カテゴリに属する語をインスタンスと呼ぶ．集合拡張とは，少数のインスタンスを入力（シード）とし，それらと同一の意味カテゴリに属する未知のインスタンスを獲得するタスクである．例えば「プリウス」や「レクサス」というシードインスタンスから,「インサイト」や「ヴィッツ」という自動車の車種を獲得する．集合拡張は語彙知識獲得の核となる技術であり，固有表現獲得，語義曖昧性解消，文書分類，クエリ解析など，自然言語処理において幅広い応用がある [10].

近年，自然言語処理では機械学習を適用した手法が広く使われ成功を収めており，集合拡張についても機械学習を用いた手法が存在する [8, 2]．これらの手法は，それぞれの意味カテゴリを識別するためのタグを付与した大規模なコーパスを訓練データとして必要とする．これに対して，人手によるタグ付けのコストを抑え，かつ特定のカテゴリに依存しない手法である半教師あり学習（ブートストラッピング）が注目されている [1, 16, 6].

ブートストラッピング [1, 16, 6] とは，シードとして与えられたインスタンスと共起するパターンを獲得し，獲得したパターンを用いて新たなインスタンスを獲得するという手続きを反復するものであり，少数のインスタンスから大規模なインスタンス集合を再帰的に獲得する手法である．代表的なものとして Espresso アルゴリズム [11] や，グラフカーネルを用いたアルゴリズム [7] が存在する．

ブートストラッピング手法の最大の問題点は，本来の意味カテゴリとは異なるインスタンスを獲得してしまうこと（意味ドリフト）である．例えば自動車の車種に関するカテゴリの集合を取得するために「プリウス」や「レクサス」などのシードを与えてブートストラッピングを行うと，反復が進むにつれて,「携帯電話」や「パソコン」などのシードとは関連の薄いインスタンスを獲得してしまう．これは一般性の高いパターンやインスタンスの多義性によって引き起こされる問題である．意味ドリフトによりシステムの出力するインスタンス集合が本来の意味カテゴリとは大きく異なってしまう場合もあり，ブートストラッピングを行う際には意味ドリフト問題への対処が求められる．

本研究では，意味ドリフトを抑制するために，比較的入手がしやすいカテゴリの上位下位関係を利用した集合拡張手法を提案する．カテゴリの抽出したインスタンスが上位カテゴリにも属するかどうかを検証することで，上位概念の性質を持ったインスタンスのみを獲得し，意味ドリフトを防ぐ．例えば図 3.1 に示すように，自動車の車種のカテゴリには自動二輪の車種を傘下に収める上位カテゴリが存在する．自動車の車種についてのインスタンスを獲得する際に，インスタンスが輸送用機器としてコーパス中で出現しているかを確認する．具体的には共通の上位カテゴリを持つ意味カテゴリ（兄弟カテゴリ）のインスタンス集合を用いて上位カテゴリのパターンを獲得し，カテゴリのインスタンスを獲得するにあたりそれをフィルタとして用いる．図 3.1 に示したように，自動車の車種のカテゴリがパターンとして「新型の X」というものを獲得していた場合，インスタンス候補として「インサイト」や「ヴィッツ」の他に「携帯電話」なども抽出してしまう．これらの候補に対し，上位カテゴリである輸送用機器のパターン,「X に乗る」とコーパス

中で共起しているかを検証する．これにより「インサイト」や「ヴィッツ」など輸送用機器としての性質を持つインスタンスのみを獲得し，意味ドリフトによって抽出された「携帯電話」などは削除される．

本論文は 5 章で構成されている．本章に続く 2 章では，意味ドリフト問題への対処を行っている関連研究について述べる．3 章ではベースラインとして用いた Espresso アルゴリズムおよび CPL アルゴリズムについて詳しく説明し，さらにこれらのアルゴリズムでの問題を解決する手法として，意味カテゴリの階層関係を活用した手法を提案する．4 章では実験の設定と結果，および結果の分析について述べる．5 章では本研究で明らかになった点，及び今後の課題について述べ，本研究のまとめを行う．

第2章　関連研究

この章ではブートストラッピングによる集合拡張手法として Pantel and Pennacchiotti による Espresso アルゴリズム [11] と Carlson らによる CPL アルゴリズム [3] を説明する．その上で，集合拡張を行う上で問題となる意味ドリフト問題について述べ，さらにこの問題に対処するための研究について記す．

2.1　Espresso アルゴリズム

Espresso アルゴリズム [11] は候補の抽出とランキングからなる．候補の抽出では，コーパス中でシードインスタンスや前回の反復で獲得したインスタンスと共起するパターン，前回の反復で獲得したパターンと共起するインスタンスを候補として抽出する．なお，本研究ではインスタンスと係り受け関係を持つ文節をパターンとして用いた．例えば次の文において「トヨタ自動車」がインスタンスであるとき，以下に挙げるパターンを抽出する．

プリウスなどを販売している トヨタ自動車 が新たに発表した.....

- X⟶ 販売している
- X⟵ 発表した

次に候補にスコアを付与し，このスコアによってランキングを行い，上位 N 個を獲得する．Espresso アルゴリズムはこのランキングのスコア付けに巧妙なスコア関数を導入することにより，一般性の高いパターンや，多義性のあるインスタンスの影響力を減らすことのできるアルゴリズムである．パターン候補 p のスコア $r_\pi(p)$，インスタンス候補 i のスコア $r_\iota(i)$ はそれぞれ以下のように与えられる．

$$r_\pi(p) = \frac{1}{|I|} \sum_{i \in I} \frac{pmi(i,p)}{\max pmi} r_\iota(i) \tag{2.1}$$

$$r_\iota(i) = \frac{1}{|P|} \sum_{p \in P} \frac{pmi(i,p)}{\max pmi} r_\pi(p) \tag{2.2}$$

$$pmi(i,p) = \log_2 \frac{|i,p|}{|i,*||*,p|} \tag{2.3}$$

ここで，P と I はそれぞれのカテゴリにおけるパターンとインスタンスの集合であり，$|P|$ と $|I|$ はパターンとインスタンスの数である．$|i,*|$ はコーパス内におけるインスタンス i の出現頻度であり，$|*,p|$ は同様にパターン p の出現頻度である．また，$|i,p|$ はパターン p とインスタンス i の共起頻度である．$\max pmi$ はカテゴリ内のすべてのインスタンスとパターンの組み合わせの間での pmi の最大値である．

ところで，上記の式により定義される pmi の値をこのまま用いると，コーパス中で出現頻度の少ないインスタンスやパターンのスコアが高くなるようなバイアスがかかってしまう．これに対処するために，実際には以下の式で定義される値と式 (2.3) との積を $pmi(i,p)$ として使用する．

$$\frac{|i,p|}{|i,p|+1} * \frac{\min(|\mathrm{i},*|,|*,\mathrm{p}|)}{\min(|\mathrm{i},*|,|*,\mathrm{p}|)+1} \tag{2.4}$$

式 (2.1)，(2.2) のスコアリング関数を用いる事により，スコアの高いパターンとよく共起するインスタンスはスコアが高く，逆にスコアの高いインスタンスとよく共起するパターンもスコアが高くなる．Espresso アルゴリズムはこのようにインスタンスとパターンのスコアを相互再帰的に定義する事により，高い適合率と再現率を実現できるようにしている．

なお，Espresso アルゴリズムでは，ランキングした結果上位 N 個以内に入っているインスタンスやパターンを次の反復のシードとしてのみ用い，システムの出力は最終反復の際のランキングにおいて上位 N 個以内に入っているインスタンスとしている．

2.2 CPL アルゴリズム

CPL アルゴリズム [3] は，候補の抽出，フィルタリング，ランキングの 3 つのフェーズにより構成される．候補の抽出では，コーパス中でシードインスタンスや既に獲得したインスタンスと共起するパターン，既に獲得したパターンと共起するインスタンスの候補を抽出する．

次に，上記のパターンやインスタンスの候補がそれぞれ 1 つの意味カテゴリに属するように，フィルタリングを行う．具体的には，あるカテゴリ x に出現しているインスタンス／パターンの候補が，別のカテゴリ y でも出現している場合，カテゴリ x での出現頻度が y での頻度の 3 倍以上であるときに限り，その候補がカテゴリ x に属することとする．例えば自動車の車種というカテゴリにおいて「マーチ」というインスタンスが出現しており，自動車の車種のカテゴリに属するパターンとの共起頻度の合計が 7000 回であったとする．このとき，他のカテゴリでも「マーチ」というインスタンスが出現しているとすると，そのカテゴリに属するパターンと「マーチ」の共起頻度の合計を 3 倍した値が 7000 を下回っているときに限り,「マーチ」は自動車の車種に属するとする．Carlson らによれば，この制約によってウェブテキストのような語の曖昧性やノイズの多い文書に対し，一般的なパターンや多義性のあるインスタンスを効果的に除外することができる．

フィルタリング後のインスタンス候補は，共起しているパターンの数でランキングする．すなわち，意味カテゴリが保有するパターンの多くと共起するインスタンスを獲得することになる．パターンの候補 p は，式 2.5 で定義される適合率に基づき，ランキングする．

$$Precision(p) = \frac{\sum_{i \in c} count(i,p)}{count(p)} \tag{2.5}$$

ここで，c はある意味カテゴリのシードインスタンスおよび獲得したインスタンスの集合，$count(i,p)$ はインスタンス i とパターン p がコーパス中で共起する回数，$count(p)$ はコーパスにおける p の頻度である．この適合率を用いることにより，意味カテゴリ c に属する確実性の高いパターンが上位にランキングされる．これらのランキングの結果，上位 N 個を新たにインスタンス，パターンとして獲得する．ただし，インスタンス／パターンはそれぞれに少なくとも 2 つ以上のパターン／シードインスタンスと獲得済みのインスタンスと共起することとする．

2.3 意味ドリフト問題

上記のブートストラッピングを行う上で最大の問題が意味ドリフト問題である．これは本来の意味カテゴリとは異なるインスタンスを獲得してしまう問題である．例えば「プリウス」や「レクサス」をシードインスタンスとする自動車の車種のカテゴリについて，ブートストラッピングを行うと，反復が進むにつれて「新型の X」や「X の性能」などの一般性の高いパターンが得られる．このパターンを用いてインスタンスを獲得すると,「携帯電話」や「パソコン」など，シードとは関連の薄いインスタンスを獲得してしまう．

また，意味ドリフトはインスタンスの多義性によっても引き起こされる．例えば，自動車メーカーに関する意味カテゴリの集合を獲得するために「スバル」や「サターン」をシードとした場合,「天王星」や「木星」など，天体を表すインスタンスを獲得してしまう．これは,「スバル」や「サターン」が自動車メーカーだけではなく天体も表すため，天体のパターンを獲得してしまうためである．

誤ったインスタンスやパターンを所持したまま反復を続けると，さらに誤ったインスタンスやパターンを獲得してしまい，最終的なシステムの出力が本来の意味カテゴリとは大きく異なってしまう可能性もある．このため，ブートストラッピングを行う際には意味ドリフト問題への対処を行う必要がある．上記の Espresso アルゴリズムや CPL アルゴリズムは一般性の高いパターンや多義性のあるインスタンスの影響を抑え，意味ドリフトへの対処を行っているが，意味ドリフトを完全に防ぐには到っていない [7]．続く節において，意味ドリフト問題を防ぐための，最新の研究について記す．

2.4 カテゴリ間の排他関係を利用した手法

意味カテゴリ間の排他関係を利用した手法として，Curran らはインスタンスとパターンはそれぞれただ 1 つのカテゴリにのみ属するという制約を導入した Mutual Exclusion Bootstrapping を考案した [4]．複数のカテゴリで出現するインスタンスやパターンは曖昧性が高く，意味ドリフトの原因と考えられるため除去することで適合率を上げる方法である．具体的にはパターンやインスタンスの候補を抽出した際に，複数カテゴリで出現しているものについては除去する．この手法では，機能語や意味ドリフトにより獲得してしまうインスタンスの集合をシードとして与えることで，これらを除去することができ，より効果的なブートストラッピングが可能になる．しかし，複数カテゴリで出現するパターンやインスタンスを除去してしまうために，再現率はかなり下がってしまう．

なお，2.2 節に記したとおり，Carlson らの考案した CPL アルゴリズム [3] も Curran らのようにカテゴリ間の排他関係を利用した手法である．

2.5 分類器を用いる手法

意味ドリフトを防ぐために，Girju らは機械学習を用いてインスタンスを分類する手法 [5] を提案した．これは高い適合率と再現率を示したが，人手によるタグ付けを行う必要があり，多大なコストが発生する．分類器を用いる際には人手による訓練データ作成のコストが大きな問題となる

が，Pennacchiotti and Pantel は人手によるタグ付けのコストを解消し，なおかつ分類器に高い性能を発揮させるための訓練データを自動で獲得する方法を提案した [12]．Wikipedia や IMDB などの大規模ソースから獲得した情報と，既存の集合拡張手法で得られた情報を組み合わせて訓練データとする．大規模ソースから得られたインスタンスと既存の集合拡張手法で得られたインスタンスのうち一致するものを正例とし，負例には近いカテゴリのインスタンス，誤りインスタンス，別のカテゴリに属しているインスタンスを用いる．近いカテゴリとは例えば俳優に対しては映画監督やアスリートなどのカテゴリである．誤りインスタンスは既存の集合拡張手法など，信頼度の低い情報源のみから得られたインスタンスである．別のカテゴリに属しているインスタンスについては，対象の意味カテゴリに近いかどうかには関係なく，単純に別のカテゴリに属しているインスタンスとしている．

Sadamitsu らは LDA によるトピック情報を用いた手法を提案している [13]．分類器の素性として，文脈情報以外にトピック情報を用い，負例の獲得や，分類器の出力についてもトピック情報を利用した手法であり，例えば負例の獲得については，LDA によるトピック情報から，対象としているカテゴリから遠いものを選んで負例にしている．Sadamitsu らはこの手法により，文脈情報を素性として用いるだけの手法よりも有意に差があることを報告している．

2.6 人手の判断を用いる手法

意味ドリフトにより獲得したインスタンスから原因を特定し，削除するアルゴリズムが Min ら [9] や Vyas ら [14] により提案されている．これらは反復の途中で獲得したインスタンスをユーザーに提示し，ユーザーからの正否判断をフィードバックするものである．具体的には獲得したインスタンスにノイズが混入していた際に，それを獲得する原因となったパターンを削除する．さらに，誤りインスタンスに似た文脈ベクトルを持っているインスタンスが存在すると，同様の誤りが発生する可能性があるため，除去した方が良いと考えられる．獲得したインスタンスのうち，ユーザーが誤りとタグをつけたインスタンスとそれ以外のインスタンスとの類似度を，文脈ベクトルを用いて計算し，誤りインスタンスとの類似度の高いインスタンスを削除することで，正確に意味カテゴリに属するインスタンスのみを残す．

2.7 シードインスタンスを洗練させる手法

意味ドリフトはインスタンスの多義性により，対象としている意味カテゴリとは関係のないパターンを獲得してしまうことによっても引き起こされる．さらに，インスタンスがどのようなパターンを獲得できるかによって，続く反復で獲得できるインスタンスが変化することから，シードインスタンスにどのようなものを選ぶかはブートストラッピングの結果への影響が非常に大きいと言える．

Vyas ら [15] はシードインスタンスをシードセットの中での頻度やクラスタリングなどいくつかの手法によって選別し，それぞれのシードセットにおける意味ドリフトの発生度合い，獲得したインスタンスの再現率の高さを調べた．結果として，人手で作成したシードはランダムに選んだものよりも性能が悪くなる可能性があることを示し，さらに，良質の結果を得るためのシードインスタンスを自動で選別する手法を提案した．これは対象の意味カテゴリに出現するパターンを

なるべく多く被覆できるようなシードインスタンスの集合を作成する手法である．既知のインスタンスを用いてシードセットを作成した際の，シードセット内のインスタンス集合が持つ文脈ベクトルの情報量の大きさを計算し，最大になるものをシードセットとする．

第3章　提案手法

2章に記した意味ドリフト問題を解決するために，本研究ではインスタンスが上位概念の性質を保有しているかどうかを検証するフィルタリング手法を提案する．具体的には共通の上位カテゴリを持つ意味カテゴリ（兄弟カテゴリ）のインスタンス集合を用いてインスタンスが上位概念の性質を保有しているかを検証できるようなパターン（上位カテゴリのパターン）を獲得し，3.1に示したように，各カテゴリのインスタンスを獲得するにあたりそれをフィルタとして用いる．すなわち，提案手法は2章に記した既存のブートストラッピング手法に，上位カテゴリのパターンを用いたフィルタリングを加えるものである．この章では上位カテゴリのパターンをどのように獲得し，どのように用いるかについて説明する．

3.1　上位カテゴリのパターンによるフィルタリング

2章に記したように，意味ドリフト問題は一般性の高いパターンや，多義性のあるインスタンスによって引き起こされる．これらの原因で獲得されるインスタンスは，多くが対象とする意味カテゴリの概念とはまったく異なるものであり，したがって，対象とする意味カテゴリの上位概念の性質も保有していないものがほとんどである．

提案手法では，インスタンスが上位概念の性質を保有しているかを検証できるようなパターンを用いて，意味ドリフトによって獲得してしまう誤りインスタンスを除去する．具体的には，インスタンス候補のランキングを行う前に，そのインスタンス候補がコーパス中で上位カテゴリのパターンと共起しているか否かを検証し，共起しているもののみをインスタンス候補として残す．

例えば図3.1に示したように，自動車の車種のカテゴリは自動二輪の車種という兄弟カテゴリを持ち，これら兄弟カテゴリは「X に乗る」や「X の燃費」などを上位カテゴリのパターンとして持つ．これらのパターンでフィルタリングを行うことにより，コーパス中でこれらと共起しない「携帯電話」や「パソコン」などは除去され，自動車の車種や自動二輪の車種に共通の上位概念を持つインスタンスのみが候補として残る．

3.2　上位カテゴリのパターンの獲得

先に記したように，上位カテゴリのパターンはインスタンスが上位概念の性質を保有しているかどうか検証できるパターンである．上位概念とは兄弟カテゴリに共通の概念であると考えられるので，上位カテゴリのパターンは兄弟カテゴリを用いて獲得する．例えば自動車の車種の上位カテゴリのパターンの獲得においては，図3.2のように，兄弟カテゴリである自動二輪の車種のカテゴリとのインスタンス集合を上位カテゴリのシードインスタンスとして，パターンを獲得する．

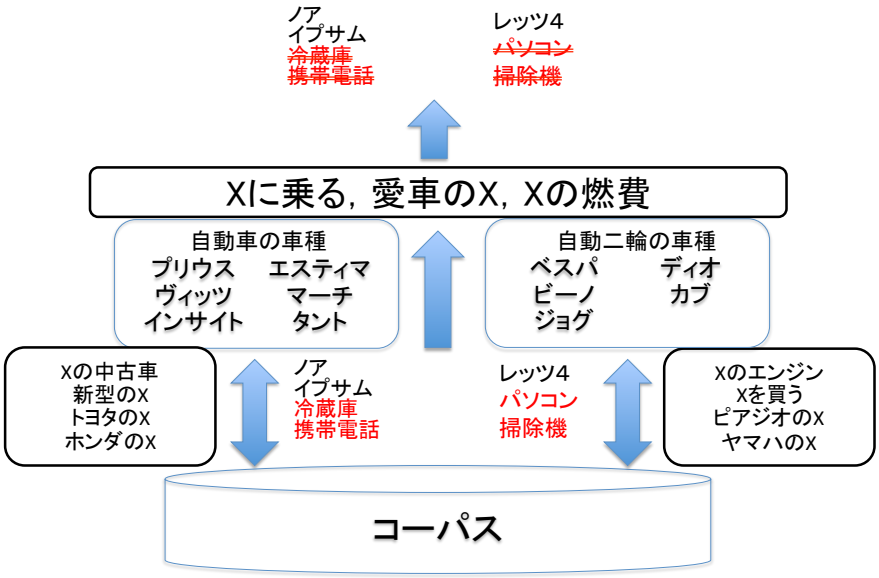

図 3.1: 提案手法による集合拡張

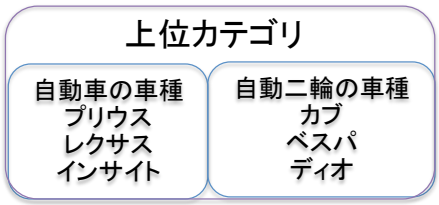

図 3.2: 上位カテゴリのパターン獲得のためのシードインスタンス

上位カテゴリのパターンは共起しているか否かによって，インスタンスを除去するべきかを検証する．したがって，兄弟カテゴリとしてグルーピングしたそれぞれのカテゴリの，全インスタンスと共起するようなパターンであることが望ましい．以下で，そのような上位カテゴリのパターンをどのように獲得するかについて記す．

上位カテゴリのパターンの獲得は候補の抽出とランキングの 2 つのフェーズからなる．候補の抽出では，上位カテゴリのシードインスタンスとコーパス中で共起するパターンを抽出する．ここでのパターンは 2 章で記したブートストラッピングで用いるパターンよりも簡素なものであり，係り受け木における親と子の区別をせず，さらに動詞と名詞に限定した．例を表 4.2 に示す．すなわち表 4.2 のような名詞や動詞が，インスタンスと係り受け関係にある文節に出現する場合，パターンとして扱う．

抽出するパターンは上位概念の性質の有無を検証できるようなパターン，すなわち兄弟カテゴリに共通のものにするため，兄弟カテゴリとしてグルーピングした複数のカテゴリのうち，2 つ以上のカテゴリで出現するものに限定する．例えば図 3.3 のように，自動車の車種と自動二輪の車種に属するインスタンスそれぞれと共起している「X のエンジン」というパターンは候補となるが，自動車の車種に属するインスタンスとしか共起していない「トヨタの X」というパターンについては，候補としない．なお，意味ドリフトを抑えるために上位カテゴリは相互に排他であるとし，複数の上位カテゴリに出現するパターンはシードインスタンスとの共起頻度が最も高いカテゴリ

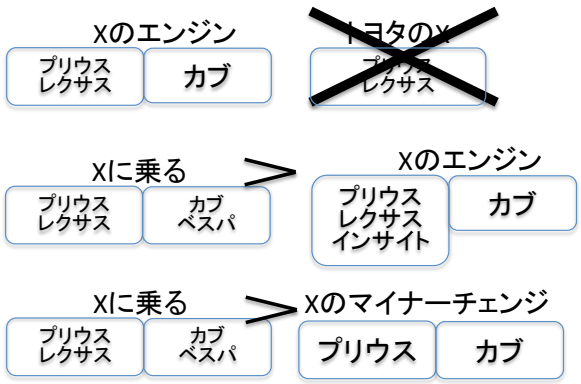

図 3.3: 上位カテゴリのパターンとして適しているパターン

に属するものとする．

抽出したパターンにスコア付けを行い，最適なものを上位カテゴリのパターンとして獲得する．ここで，上位カテゴリのパターンが持つべき特徴について，詳しく考える．ブートストラッピングにおいて，個々の意味カテゴリに属しているかの認定は，下位のカテゴリのパターンが担っている．また，上位カテゴリのパターンによるフィルタを通過できないインスタンスは削除されてしまうため，フィルタは下位のカテゴリを網羅的にカバーし，正解インスタンスを漏らさず通過させるようなものが望ましい．さらに上位カテゴリのパターンは，傘下のカテゴリのインスタンス候補が兄弟カテゴリに共通の概念を有するか検証できるパターンである必要があるため，特定のカテゴリに偏って出現しているものは適当でない．すなわち，上位カテゴリのシードインスタンスとより多く共起するもの，共起するインスタンスが各カテゴリに均等に分布しているものを選択することが望ましい．例えば図 3.3 のように，自動車の車種と自動二輪の車種の上位カテゴリのパターンとしては,「X のエンジン」や「X のマイナーチェンジ」というパターンよりも,「X に乗る」というパターンがより適当であると考えられる．このようなパターンを獲得するために，以下の式 3.1 によりスコア付けを行う．式 3.1 は，上位カテゴリ C のパターン候補 p に対して，獲得できるインスタンスのエントロピーと再現率という観点でスコア付けするものである．

$$Score(C,p) = Entropy(C,p) * Recall(C,p) \tag{3.1}$$

$$Entropy(C,p) = -\sum_{c \subset C} P_c(p) \log_{|C|} P_c(p) \tag{3.2}$$

$$Recall(C,p) = \frac{\sum_{c \subset C} cooccur(p,c)}{\sum_{c \subset C} |I_{s_c}|} \tag{3.3}$$

$$P_c(p) = \frac{cooccur(p,c)}{\sum_{c \subset C} cooccur(p,c)} \tag{3.4}$$

$$cooccur(p,c) = |I_{s_c} \cap I_{p_c}| \tag{3.5}$$

ただし，$|I_{s_c}|$ はカテゴリ c のシードインスタンス数，$|I_{p_c}|$ はパターン p の共起するカテゴリ c のインスタンス数である．よって，$|I_{p_c} \cap I_{s_c}|$ はパターン p の共起するカテゴリ c のシードインスタンス数である．C は兄弟カテゴリの集合であり，$|C|$ は兄弟カテゴリの数である．$Entropy(C,p)$ によって，上位カテゴリに属する各カテゴリのそれぞれに，パターンがどの程度の偏りで出現し

ているかを計算することができる．また，$Recall(C,p)$ によって，上位カテゴリのシードインスタンスを，パターンがどの程度網羅的にカバーしているかを計算することができる．したがって，$Entropy(C,p)$ と $Recall(C,p)$ の積によって得られる $Score(C,p)$ によって，パターンが上位カテゴリのパターンとしてどの程度適切であるかを計算することが可能である．

各上位カテゴリの候補パターン p について $Score(p)$ を計算し，上位 N 個をパターンとする．なお N の値は獲得したパターンによるシードインスタンスのカバー率により調整する．ここではシードインスタンスのカバー率が 100 %となるまで，あるいはパターンの数が 15 個になるまでとした．

第4章　実験

4.1　実験設定

実験にはコーパスとして，ウェブページ 1 億 1 千万文書を KNP で解析したものを用いた．ただし，計算時間が膨大になることを防ぐため，パターン，インスタンス共に出現頻度 2 以下のものは取り除いた．実験に用いた 41 個の意味カテゴリは表 4.2 の左側に記したとおりである．それぞれのカテゴリは必ずただ 1 つの上位カテゴリを持ち，上位カテゴリには 2 つ以上のカテゴリが属するものとした．表 4.2 では上位カテゴリが共通のカテゴリ毎に色を分けて記している．また，シードインスタンスは各カテゴリで 15 個とした．各カテゴリのシードインスタンスについては付録 A を参照されたい．シードインスタンスとカテゴリ間の上位下位関係については，隅田らのツール [17] を Wikipedia に適用し，その出力結果からノイズを人手で除去したものを利用した．

Carlson らの CPL アルゴリズム，ならびに Pantel and Pennacchiotti による Espresso アルゴリズムに排他制約を加えた手法をベースラインとし，それぞれに提案手法によるフィルタリングを加えた手法と比較する．

Espresso アルゴリズムに加える排他制約として，複数のカテゴリにおいて出現するパターン，インスタンスについては候補をランキングした際に上位に存在するカテゴリにのみ属するとする．例えば,「X⟵ マフラー」というパターンが自動車の車種と自動二輪の車種というカテゴリにおいて出現していたとする．ランキングの結果，このパターンは自動車の車種では 13 位に，自動二輪の車種では 4 位であったとすると，自動二輪の車種に属するものとし，自動車の車種からは除去する．この制約により，排他関係を利用しつつ，スコアの高いインスタンスやパターンについては獲得し，ブートストラッピングに利用する事ができると考えられる．

なお，CPL アルゴリズムにおける各反復での獲得インスタンス／パターンはそれぞれのカテゴリについて，ランキングの上位 15 個とする．Espresso アルゴリズムにおける次の反復のシードとするインスタンス／パターンの数についても初期値を 15 個とし，反復毎に各カテゴリで 15 個ずつ増えていくものとする．

4.2　評価尺度

評価は適合率 (Precision) と獲得インスタンス数により評価する．適合率は以下の式 (4.1) により与えられる．

$$\text{適合率} = \frac{\text{獲得した正解インスタンス数}}{\text{獲得インスタンス数}} \tag{4.1}$$

未知のインスタンスを獲得することを目的とするため，再現率は正しく求めることができない．よって獲得インスタンス数に対する適合率を比較することでシステムの性能を測定する．ブート

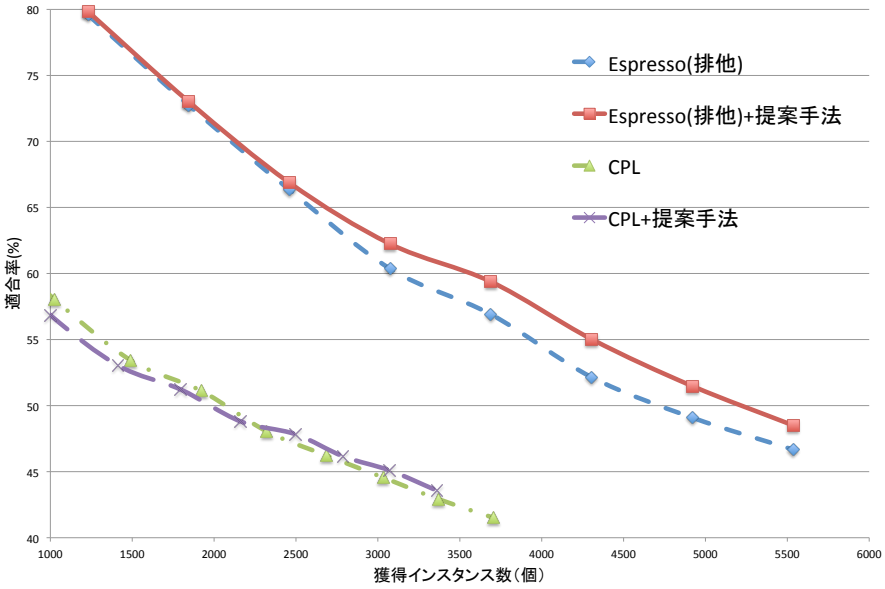

図 4.1: 各手法における獲得インスタンスと適合率

ストラッピングの反復毎にシステムが獲得した全インスタンスを評価する．インスタンスの正否は人手により判定する．

4.3 実験結果

提案手法とベースラインでブートストラッピングの反復を 8 回行った際の，各反復における全カテゴリでの獲得インスタンス数とその適合率を図 4.1 に示す．Espresso(排他)，CPL はベースラインのアルゴリズムである．これらに提案手法である上位カテゴリのパターンによるフィルタを加えたものがそれぞれ Espresso(排他)+提案手法，CPL+提案手法である．図 4.1 より，獲得インスタンスが増えた際に提案手法を用いた Espresso(排他)+提案手法，CPL+提案手法はそれぞれ Espresso(排他)，CPL よりも適合率が上がっている．このことから，提案手法を用いることにより意味ドリフトを防ぐことが可能であると言える．提案手法を用いた際の適合率の上昇幅には CPL と Espresso(排他) で差があるが，これについては続く 4.4 節で議論する．

図 4.1 のとおり，Espresso(排他) は CPL よりも適合率が高く，また，ベースラインと提案手法において各反復で獲得するインスタンス数が等しい．以下では Espresso(排他) と Espresso(排他)+提案手法について詳しく調べる．Espresso(排他) と提案手法によるフィルタリングを用いた Espresso(排他)+提案手法について，各カテゴリで 90 個のインスタンスを獲得したときの，神社，寺，新聞，雑誌の 4 つのカテゴリにおけるスコアの上位 15 個のインスタンスを表 4.1 に示す．新聞と雑誌，神社と寺はそれぞれ上位カテゴリが共通のカテゴリであり，それぞれにおける上位カテゴリのパターンも表 4.1 に記した．表 4.1 においてはベースラインで獲得した正解インスタンスは黒字で，ベースラインで獲得した誤りインスタンスは灰字で記し，提案手法を用いることにより獲得できたインスタンスのうち，正解は赤字で，誤りは青字で記した．なお，誤りインスタンスには色に加えて線を引いた．表 4.1 には獲得インスタンスのうち，スコアの上位 15 個を記している．すなわち，カテゴリに属するインスタンスとして確信度の高いものである．Espresso(排他)

手法	カテゴリ	インスタンス	上位カテゴリのパターン
Espresso (排他)	神社	太宰府天満宮, 明治神宮, 八幡宮, 伊勢神宮, 天満宮, 出雲大社, 鶴岡八幡宮, 春日大社, 八坂神社, 靖国神社, 神田明神, 愛宕神社, ~~神社, 成田山, 浅草寺~~	
	寺	祇王寺, 徳林庵, ~~木島坐天照御魂神社, 許波多神社, 京都府立堂本印象美術館, 宇治上林記念館, あだしのまゆ村, 袋屋醤油店, 京都ハリストス正教会, 今昔西村, 薫玉堂, 伊根湾めぐり遊覧船, カトリック宮津教会, 琵琶湖疎水記念館, 大原観光保勝会~~	
	新聞	日経新聞, 日本経済新聞, 朝日新聞, 日経新聞朝刊, 米紙ウォールストリートジャーナル, 米紙ワシントンポスト, 地元紙, 米紙ニューヨークタイムズ, 読売, ニューヨークタイムズ紙, ~~日本経済新聞朝刊, 朝刊, 朝日新聞朝刊, 英科学誌ネイチャー, 米科学誌サイエンス~~	
	雑誌	CQ誌, 月刊タウン情報トクシマ, Mart, CanCam, ViVi, Oggi, AneCan, LaLa, POTATO, JJ, MISS, VOCE, VERY, ESSE, ~~市報松江~~	
Espresso (排他) +提案手法	神社	明治神宮, 伊勢神宮, 太宰府天満宮, 八幡宮, 鶴岡八幡宮, 出雲大社, 八坂神社, 春日大社, 天満宮, 靖国神社, 神田明神, 愛宕神社, 厳島神社, ~~神社, 成田山~~	境内 梅 初詣
	寺	南禅寺, 大徳寺, 知恩院, 妙心寺, 六波羅蜜寺, 相国寺, 常寂光寺, 赤山禅院, 来迎院, 金蔵寺, ~~寺, お寺, 下鴨神社, 今宮神社, 平野神社~~	
	新聞	中日新聞, 日経新聞, 日本経済新聞, 朝日新聞, 日経新聞朝刊, 米紙ワシントンポスト, 米紙ニューヨークタイムズ, 米紙ウォールストリートジャーナル, 地元紙, ニューヨークタイムズ紙, 読売, ~~日本経済新聞朝刊, 朝刊, 朝日新聞朝刊,~~ 読売新聞朝刊	載る 記事 掲載
	雑誌	CQ誌, CanCam, ViVi, ESSE, VOCE, LaLa, MISS, POTATO, VERY, Oggi, JJ, dancyu, アニメディア, コンプティーク, 中央公論	

表 4.1: Espresso と提案手法によるフィルタリングを用いた手法の獲得インスタンス（90 個中上位 15 個）

では寺や新聞のカテゴリにおいて誤りインスタンスが多い．特に寺カテゴリでは深刻な意味ドリフトが発生していると考えられる．これに比べ Espresso(排他)+提案手法は誤りインスタンスの数が表 4.1 に示したすべてのカテゴリにおいて減少しており，提案手法による，上位カテゴリのパターンを用いたフィルタリングが有効であることがわかる．

さらに，各カテゴリで 90 個のインスタンスを獲得した際の，Espresso(排他) と Espresso(排他)+提案手法の適合率を表 4.2 に示す．表 4.2 では，上位カテゴリが共通のカテゴリ毎に色を分けて記し，それぞれで取得できた上位カテゴリのパターンとそのスコアを，スコアの高いものから 3 つずつ記した．表 4.2 より，Espresso(排他) と Espresso(排他)+提案手法において，多くのカテゴリでは適合率が変化しないか上昇するが，いくつかのカテゴリでは下降していることがわかる．これについては続く 4.4 節で詳しく議論する．

4.4 分析

表 4.1 について，Espresso(排他) では寺カテゴリにおいて「あだしのまゆ村」や「袋屋醤油店」など本来の意味カテゴリとはまったく関係のないインスタンスを取得してしまっている．このようなインスタンスを Espresso(排他)+提案手法では除去できており，提案手法によって上位カテゴリの性質を保有しないインスタンスは正しく除去できていると言える．すなわち，表 4.1 に記した上位カテゴリのパターンによって，インスタンスが上位カテゴリの性質を有しているかどうかを

判別できており，提案手法は意味ドリフトの抑制に効果がある．しかしながら，表 4.1 のとおり，上位カテゴリが共通のカテゴリがインスタンス候補となっているときに誤って獲得してしまう問題があり，カテゴリ同士の境界をどのように定義するか，検討が必要である．

表 4.2 に示したように，いくつかのカテゴリでは提案手法を用いた際に適合率が下がってしまっている．適合率が下がる場合は大きく分けて，次の 2 つに分類されると考えられる．

1. ベースラインでの適合率が高い

2. 上位カテゴリのパターンのスコアが低い

1. には映画監督や日本の都市などのカテゴリが当てはまる．1. のケースの場合，ベースラインで意味ドリフトが発生していないにも関わらず，フィルタを用いることで正解インスタンスをフィルタリングしてしまっていることにより，適合率が減少している．この問題はすべてのカテゴリのすべてのインスタンスにフィルタを適用しているために発生すると考えられる．インスタンスやパターンの確信度などから提案手法によるフィルタを用いるか否かを各カテゴリについて判断し，意味ドリフトが発生していないカテゴリではフィルタリングを行わないようにしなければならない．

また，フィルタに用いる上位カテゴリのパターンはブートストラッピングの前に決定され，更新されることがない．正解インスタンスをフィルタリングしてしまっているのはこれが原因である可能性も考えられる．すなわち，現状の手法では収束が早く，再現率が十分でない可能性がある．これを解決するためには，ブートストラッピングの反復を続けた際に，上位カテゴリのパターンを更新する必要がある．

2. は上位のカテゴリが共通のカテゴリすべてで適合率が減少している場合であり，自動車メーカーと医薬品メーカーや細菌とウイルスのカテゴリなどが当てはまる．提案手法では上位カテゴリが共通のカテゴリ間で共通に用いられるパターンをフィルタとして使用するようにしており，スコアはパターンの共通性の指標となる値である．したがって，得られたパターンのスコアが低いということは，カテゴリ間の共通性がもともと少ないということである．共通性の少ないカテゴリ同士が組となってしまっているのは，Wikipedia の情報を元にしたためであろう．カテゴリがどの上位カテゴリに属するか，すなわち，どのカテゴリ同士を組にするかを定める手法についても考える必要がある．

また，ベースラインに用いる手法についてであるが，図 4.1 のとおり，CPL アルゴリズムと Espresso(排他) アルゴリズムとでは適合率にかなりの差がある．これはインスタンス／パターンを獲得する際のスコア関数の違いによるものであると考えられる．CPL ではそれぞれの意味カテゴリに属する個々のインスタンス／パターンは確実にそのカテゴリに属するとし，パターン／インスタンスを獲得する際にはインスタンス／パターンとの共起頻度と種類のみが重要となる．すなわち，一般性の高いパターンや多義性のあるインスタンスを獲得した際に，意味ドリフトが非常に発生しやすい．これに対し，Espresso(排他) では意味カテゴリに属する個々のインスタンス／パターンに確信度が付与されており，確信度の高いインスタンス／パターンとどの程度共起しているかでスコアが決定する．これにより，Espresso(排他) では一般性の高いパターンや多義性のあるインスタンスを獲得しても意味ドリフトが発生しづらくなっている．

CPL アルゴリズムと Espresso(排他) アルゴリズムとにはまた，獲得できるインスタンス数に違いがあり，提案手法である上位カテゴリのパターンによるフィルタを用いた際の適合率の上昇幅に

も差が見られる．これは CPL でのカテゴリ間における排他制約が強いためであると考えられる．CPL は上記にあるように，一般性の高いパターンや多義性のあるインスタンスを獲得した際に，意味ドリフトが発生しやすい．これを防ぐために，CPL ではそれぞれの意味カテゴリで出現するインスタンス／パターンが他のカテゴリでの出現頻度に比べ，突出して高くなければ候補から除去するという排他制約がある．この排他制約で大量の候補を除去した結果，獲得できるインスタンス数が Espresso(排他) と比べ減少しているのだと考えられる．また，この排他制約を用いても候補となっているインスタンスは，対象としている意味カテゴリに属していなくとも，関連は非常に高いものである．したがって上位カテゴリのパターンを用いたフィルタによっても除去が難しいため，Espresso(排他) と比べ，CPL では提案手法による適合率の上昇が小さいのだと考えられる．

この問題はベースラインとしている手法による部分もあるが，上位カテゴリのパターンをフィルタとしてしか用いていないことにも一因がある．先には，カテゴリ毎に上位カテゴリのパターンを用いたフィルタを適用するか否かを判断する必要があると記したが，ランキングする際のスコアに上位カテゴリの情報を用いるなど，個々のカテゴリのパターンと組み合わせた使用法についても検討をする必要があるだろう．

カテゴリ	Espresso(排他)の適合率(%)	Espresso(排他)+提案手法の適合率(%)	上位カテゴリのパターン	上位カテゴリのパターンのスコア
神社	73.33	76.67	境内	0.9658
			梅	0.5946
寺	37.78	62.22	初詣	0.5266
映画監督	97.78	74.44	原作	0.6235
漫画家	86.67	91.11	代表作	0.5832
小説家	93.33	94.44	傑作	0.3167
自動車メーカー	62.22	47.78	株価情報	0.1837
			武田薬品	0.1333
医薬品メーカー	5.56	3.33	株主総会	0.1333
日本の都市	97.78	95.56	住む	1.0000
アメリカ合衆国の都市	34.44	36.67	行く	1.0000
中華人民共和国の都市	58.89	61.11	向かう	0.9319
感染症	47.78	47.78	病気	1.0000
			治療	0.9658
精神疾患	45.56	45.56	症状	0.9658
ボードゲーム	23.33	23.33	やる	0.9092
コンピュータゲーム	98.89	98.89	ゲーム	0.8651
カードゲーム	61.11	62.22	遊べる	0.4434
自動車の車種	95.56	95.56	乗る	0.7572
			愛車	0.5510
自動二輪の車種	82.22	92.22	あなた	0.5409
アジアの国	33.33	33.33	日本	1.0000
アフリカの国	45.56	45.56	いう	1.0000
ヨーロッパの国	48.89	48.89	国	1.0000
湖	21.11	21.11	ほとり	0.8518
			望む	0.7146
池	0	0	畔	0.6618
島	66.67	61.11	流れる	0.5757
山地	96.67	92.22	釣る	0.4412
河川	95.56	95.56	海	0.3749
ラジオ局	61.11	61.11	番組	0.9658
			放送	0.8630
テレビ局	67.78	67.78	アナウンサー	0.8630
鉄道駅	100	100	降り立つ	0.6989
			近く	0.6042
空港	37.78	43.33	羽田空港	0.5090
元素	20	20	含む	1.0000
			量	0.9299
化合物	41.11	41.11	成分	0.8518
美術館	37.78	17.78	外観	0.1618
			前庭	0.1203
劇場	50	24.44	閉館	0.1082
俳優	94.44	94.44	画像	0.8920
			動画	0.8518
コメディアン	98.89	97.78	出演	0.8324
細菌	45.56	40	細菌	0.4585
			菌	0.4460
ウイルス	66.67	61.11	大腸菌	0.4183
新聞	48.89	63.33	載る	0.8920
			記事	0.7635
雑誌	44.44	97.78	掲載	0.5698
出版社	23.33	96.67	発売元	0.6473
			おなじみ	0.2014
レコード会社	47.78	48.89	製造元	0.1656

表 4.2: インスタンスを 90 個獲得したときの Espresso と提案手法によるフィルタリングを用いた手法の適合率と上位カテゴリのパターン

第5章　まとめ

本論文ではカテゴリ間の関係を利用した集合拡張法として，カテゴリの上位下位関係を用いることにより，意味ドリフトを抑える手法を提案した．評価実験の結果，最新の研究である，巧妙なスコア関数によって高い適合率を保ちつつ再現率を上げる Pantel and Pennacchiotti による Espresso アルゴリズムに排他制約を加えたものや，カテゴリ間の関係を用いることで適合率を上げる Carlson らの CPL アルゴリズムよりも，精度を向上させることができた．

しかしながら，4.4 節で記したように，本論文で提案した手法は組にする兄弟カテゴリによっては適合率が下がってしまうため，類似性の高いカテゴリ同士を選択しなければならない．また提案手法では上位カテゴリのパターンによるフィルタをすべてのカテゴリのすべてのインスタンスに適用するため，意味ドリフトの発生していないカテゴリでは逆に適合率が下がってしまう可能性がある．さらに，フィルタとして用いるため，インスタンス候補が減少するのみで，再現率が下がってしまう可能性もある．これに対処するために，ブートストラッピングの反復を繰り返したときの上位カテゴリのパターンの更新方法や，上位カテゴリのパターンをフィルタとしてではなく，インスタンス獲得時のランキングに用いる手法などを検討する必要がある．今後はそのような，個々のカテゴリの情報と上位カテゴリの情報とを組み合わせ，より適合率と再現率の高い集合拡張法を実現できるよう，研究を進めていきたい．

さらに，上位カテゴリの情報が集合拡張において有用であることが明らかとなったが，上位カテゴリにも上位カテゴリがあり，また，個々のカテゴリには下位カテゴリも存在する．将来的にはそれらの情報を元にインスタンスが属するどのカテゴリに属するかの判断を行えるシステムを構築したい．

謝 辞

本研究を進めるにあたり，ご指導を頂いた乾健太郎教授，岡崎直観准教授に感謝いたします.

ツールの解説およびデータを快く提供していただいた NHK 放送技術研究所の山田一郎氏に感謝いたします.

日常の議論を通じて多くの知識や示唆を頂いた乾・岡崎研究室の皆様に感謝いたします.

参 考 文 献

[1] S. Abney. Understanding the Yarowsky Algorithm. *Computational Linguistics*, Vol. 30, No. 3, 2004.

[2] Razvan Bunescu and Raymond Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, 2004.

[3] 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 (WSDM 2010)*, 2010.

[4] James R. Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Pacific Association for Computational Linguistics*, 2007.

[5] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Comput. Linguist.*, 2006.

[6] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*. Association for Computational Linguistics, 1992.

[7] Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[8] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003.

[9] Bonan Min and Ralph Grishman. Fine-grained entity set refinement with user feedback. In *Proceedings of the RANLP 2011 Workshop on Information Extraction and Knowledge Acquisition*, 2011.

[10] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, 2009.

[11] 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 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[12] Marco Pennacchiotti and Patrick Pantel. Automatically building training examples for entity extraction. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011.

[13] Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura, and Genichiro Kikui. Entity set expansion using topic information. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*. Association for Computational Linguistics, 2011.

[14] Vishnu Vyas and Patrick Pantel. Semi-automatic entity set refinement. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

[15] 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*, 2009.

[16] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 1995.

[17] 隅田飛鳥, 吉永直樹, 鳥澤健太郎. Wikipedia の記事構造からの上位下位関係抽出. 自然言語処理 = Journal of natural language processing, Vol. 16, No. 3, 2009.

付 録A　シードインスタンス

本実験に用いた各カテゴリのシードインスタンスを以下に示す.

カテゴリ	シードインスタンス
神社	靖国神社,出雲大社,明治神宮,八坂神社,厳島神社,鶴岡八幡宮,北野天満宮,春日大社,日光東照宮,花園神社,太宰府天満宮,熊野神社,熱田神宮,氷川神社,諏訪神社
寺	清水寺,菩提寺,観音寺,護国寺,西大寺,石山寺,大徳寺,成田山新勝寺,鞍馬寺,興福寺,万福寺,広隆寺,大石寺,定禅寺,円覚寺
映画監督	三谷幸喜,黒澤明,押井守,石井竜也,陣内孝則,三船敏郎,田辺誠一,小津安二郎,大友克洋,堤幸彦,蜷川実花,三池崇史,岩松了,庵野秀明,市川崑
漫画家	水木しげる,尾田栄一郎,ゆでたまご,赤塚不二夫,西原理恵子,原哲夫,浦沢直樹,藤子不二雄,松本零士,井上雄彦,永井豪,高橋留美子,萩尾望都,長谷川潤,一条ゆかり
小説家	東野圭吾,伊坂幸太郎,宮部みゆき,太宰治,夏目漱石,乙一,石田衣良,万城目学,奥田英朗,芥川龍之介,浅田次郎,松本清張,森見登美彦,吉田修一,琴音
自動車メーカー	フェラーリ,ポルシェ,トヨタ自動車,アウディ,日産自動車,クライスラー,フォルクスワーゲン,ランボルギーニ,オペル,ノーブル,モーリス,現代自動車,ランドローバー,日野自動車,ダイハツ工業
医薬品メーカー	エーザイ,三共,久光製薬,田辺三菱製薬,ホーユー,バクスター,近江兄弟社,山之内製薬,三菱ウェルファーマ,ミドリ十字,キッセイ薬品工業,千寿製薬,日本ケミカルリサーチ,扶桑薬品工業,キリンファーマ
日本の都市	東京,大阪,京都,神戸,奈良,博多,水戸,明石,府中,大分市,港町,金沢市,長野市,長崎市,熊本市
アメリカ合衆国の都市	コロンビア,ボストン,ホノルル,フェニックス,アトランタ,ピア,ダラス,サンノゼ,オーガスタ,デンバー,デトロイト,オリンピア,サンタフェ,オースティン,リッチモンド
中華人民共和国の都市	香港,上海,北京,マカオ,大連,広州,青島,天津,南京,深セン,蘇州,北京市,成都,杭州,西安
感染症	インフルエンザ,肺炎,豚インフルエンザ,結核,化膿,イボ,結膜炎,胃腸炎,帯状疱疹,ハンセン病,手足口病,コレラ,敗血症,梅毒,髄膜炎
精神疾患	パニック障害,統合失調症,自律神経失調症,摂食障害,恐怖症,神経症,強迫性障害,適応障害,双極性障害,神経衰弱,人格障害,ギャンブル依存症,記憶障害,気分障害,解離性障害
ボードゲーム	将棋,囲碁,麻雀,クラウン,オセロ,チェス,すごろく,人生ゲーム,シークエンス,モノポリー,ドミニオン,野球盤,アクワイア,バックギャモン,大局観
コンピュータゲーム	モンスターハンター,ロックマン,スーパーマリオ,雷電,鬼武者,ボンバーマン,メタルギアソリッド,デビルメイクライ,悪魔城ドラキュラ,ベヨネッタ,スペランカー,エースコンバット,ドンキーコング,ファンタシースターユニバース,塊魂
カードゲーム	タロット,トランプ,ブラックジャック,百人一首,花札,かるた,セブンブリッジ,ヴァイスシュヴァルツ,デュエル・マスターズ,ソリティア,大富豪,遊戯王,めんこ,ドミニオン,アクエリアンエイジ
自動車の車種	マーチ,プリウス,チェリー,ノア,デュエット,モコ,キューブ,キャミ,スカイライン,ピノ,エスティマ,プログレ,カローラ,シルビア,アルファード
自動二輪の車種	ビーノ,ドラッグスター,マジェスティ,ロードレーサー,チョイノリ,レッツ,ヴェクスター,アヴェニス,ジョグ,シルバーウイング,ボクスン,ラクーン,カブ,ズーク,ディオ
アジアの国	日本,中国,台湾,インド,ベトナム,シンガポール,イラク,フィリピン,イラン,インドネシア,カンボジア,トルコ,イスラエル,マレーシア,パキスタン
アフリカの国	エジプト,モロッコ,ケニア,カメルーン,ガーナ,ナイジェリア,エチオピア,ソマリア,ジンバブエ,タンザニア,スーダン,チュニジア,ウガンダ,リビア,アルジェリア
ヨーロッパの国	フランス,ドイツ,イタリア,イギリス,スペイン,オランダ,スイス,チェコ,ギリシャ,スウェーデン,デンマーク,ポルトガル,フィンランド,アイルランド,ベルギー
湖	琵琶湖,河口湖,浜名湖,洞爺湖,諏訪湖,霞ヶ浦,十和田湖,摩周湖,大沼,西湖,富士五湖,宍道湖,阿寒湖,田沢湖,白樺湖
池	五色沼,大正池,洗足池,明神池,猿沢池,金鱗湖,弁天池,雲場池,入鹿池,満濃池,巨椋池,夜叉ヶ池,白馬大池,深泥池,田代池
島	石垣島,グアム,バリ島,鹿島,宮古島,屋久島,淡路島,奄美大島,竹島,対馬,西表島,八丈島,硫黄島,サムイ島,松島
山地	白神山地,中国山地,紀伊山地,四国山地,九州山地,丹沢山地,日本アルプス,六甲山地,関東山地,阿武隈高地,秩父山地,奥羽山脈,夕張山地,石狩山地,養老山地
河川	石川,小川,旭川,荒川,立川,多摩川,四万十川,市川,鴨川,江戸川,道頓堀,利根川,北川,淀川,長良川

カテゴリ	シードインスタンス
ラジオ局	ニッポン放送,文化放送,毎日放送,朝日放送,ニコニコ生放送,東北放送,ラジオ関西,ラジオ大阪,東海ラジオ,ラジオ日本,四国放送,中部日本放送,エフエム東京,音泉,九州朝日放送
テレビ局	フジテレビ,テレビ朝日,日本テレビ,テレビ東京,関西テレビ,読売テレビ,テレビ大阪,東海テレビ,テレビ愛知,中京テレビ,日本放送協会,ディスカバリーチャンネル,福島テレビ,仙台放送,名古屋テレビ
鉄道駅	東京駅,京都駅,新宿駅,名古屋駅,横浜駅,大阪駅,渋谷駅,博多駅,上野駅,品川駅,新橋駅,札幌駅,仙台駅,池袋駅,広島駅
空港	福岡空港,鹿児島空港,宮崎空港,旭川空港,熊本空港,長崎空港,北九州空港,東京国際空港,高知空港,スキポール空港,八尾空港,バンコク国際空港,成田空港,フランクフルト空港,神津島空港
元素	銀,鉄,ミネラル,酸素,カルシウム,リン,銅,ゲルマニウム,水素,マグネシウム,亜鉛,鉛,カリウム,チタン,塩素
化合物	二酸化炭素,塩酸,硫酸,リン酸,炭酸カルシウム,二酸化ケイ素,硝酸,硝酸塩,クロロゲン酸,アリシン,亜硝酸,塩化ナトリウム,リン酸塩,ヒノキチオール,キシレン
美術館	名古屋市美術館,国立新美術館,東京国立博物館,愛知県美術館,九州国立博物館,国立西洋美術館,ルーヴル美術館,東京都美術館,東京都現代美術館,横浜美術館,東京都写真美術館,オルセー美術館,サントリー美術館,メトロポリタン美術館,大原美術館
劇場	中日劇場,中座,帝国劇場,国立劇場,新橋演舞場,日生劇場,南座,東京芸術劇場,宝塚大劇場,青山劇場,東京宝塚劇場,グローブ座,シアターコクーン,新宿コマ劇場,東京グローブ座
俳優	亀梨和也,赤西仁,小栗旬,ユチョン,木村拓哉,二宮和也,瑛太,北川悠仁,オダギリジョー,妻夫木聡,横山裕,向井理,藤原竜也,山田孝之,市原隼人
コメディアン	松本人志,タモリ,島田紳助,ビートたけし,志村けん,陣内智則,友近,鳥居みゆき,明石家さんま,岡村隆史,桜塚やっくん,狩野英孝,小島よしお,千原ジュニア,稲川淳二
細菌	乳酸菌,ビフィズス菌,大腸菌,腸内細菌,クラミジア,納豆菌,黄色ブドウ球菌,結核菌,ボツリヌス菌,マイコプラズマ,ブドウ球菌,サルモネラ,インフルエンザ菌,緑膿菌,酢酸菌
ウイルス	インフルエンザ,ノロウイルス,インフルエンザウイルス,天然痘,ロタウイルス,ヒトパピローマウイルス,ファージ,コロナウイルス,エンテロウイルス,麻疹ウイルス,ヒト免疫不全ウイルス,風疹ウイルス,サイトメガロウイルス,コイヘルペスウイルス,タバコモザイクウイルス
新聞	読売新聞,朝日新聞,毎日新聞,中日新聞,産経新聞,中央日報,下野新聞,神戸新聞,中国新聞,京都新聞,沖縄タイムス,山陽新聞,新潟日報,佐賀新聞,信濃毎日新聞
雑誌	すばる,週刊金曜日,週刊朝日,週刊現代,ファウスト,ガロ,アニメージュ,アニメディア,ディズニーファン,メフィスト,日経サイエンス,ガンダムエース,ニューズウィーク,アサヒカメラ,メガミマガジン
出版社	リクルート,東洋経済新報社,新潮社,ダイヤモンド社,旺文社,岩波書店,文藝春秋,技術評論社,白泉社,宝島社,一迅社,幻冬舎,徳間書店,双葉社,竹書房
レコード会社	キングレコード,ユニバーサルミュージック,ビクターエンタテインメント,ランティス,バップ,ディスクユニオン,ミディ,ハミングバード,エイベックス・マーケティング,ワーナーミュージック・ジャパン,ビーイング,日本クラウン,日本コロムビア,ソニー・ミュージックエンタテインメント,アップフロントワークス