意味カテゴリの階層関係を活用した集合拡張

高瀬翔† 岡崎直観†‡ 乾健太郎†
東北大学† 科学技術振興機構　さきがけ‡
{takase, okazaki, inui}@ecei.tohoku.ac.jp

1 はじめに

集合拡張とは，意味カテゴリに属する少数のインスタンス（シード）から，そ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 [5]．

本研究では，人手による資源作成のコストを抑え，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）に着目する．ブートストラッピングとは，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る．代表的なものとして *Espresso* [6] や，グラフカーネルを用いたアルゴリズム [3] が存在する．

1.1 意味ドリフトへの対処

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「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．このパターンと共起する名詞をインスタンスとして獲得すると，「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，意味ドリフトはシード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である．

シードインスタンスの曖昧性の問題に対処するため，Vyas ら [8] はシードインスタンスを選別することにより，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，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た．Min ら [4] や Vyas ら [7] により，反復の途中で獲得したインスタンスとパターンをユーザに提示し，ユーザーからの正否判断を効率よくフィードバックするアルゴリズムも提案されている．これらの研究では，獲得したインスタンスにノイズが混入した際に，それを獲得する原因を作ったパターンを見つけ出し，インスタンスの確信度を再計算している．

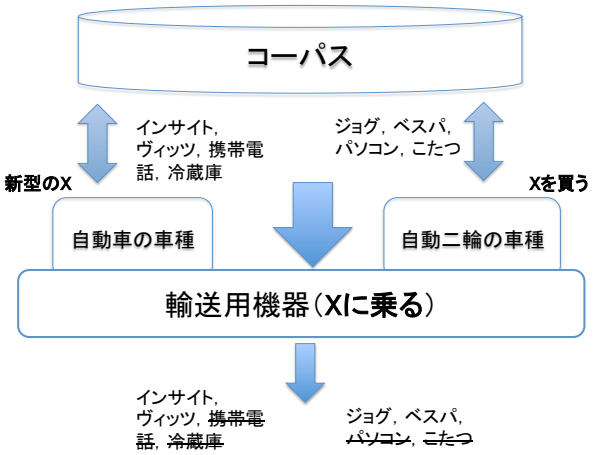

図 1: 提案手法による集合拡張

ユーザーからのフィードバックを得ずに，シードインスタンスが属する意味カテゴリ間の関係を利用して意味ドリフトを防ぐ手法も提案されている．Curran ら [2] は，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，意味ドリフトの要因となるので，インスタンスとパターンはそれぞれただ 1 つのカテゴリのみに属するという制約を導入した Mutual Exclusion Bootstrapping を考案した．Carlson らはさらに，複数カテゴリでのブートストラッピングを並行に実行し，複数カテゴリで出現しているインスタンスやパターンを単一のカテゴリに所属させる CPL アルゴリズムを提案した [1]．これはどのカテゴリでも出現頻度の変わらないような曖昧なものを除去し，確信度の高いもの（あるカテゴリでの出現頻度が突出しているもの）はそのカテゴリに属すると定める手法である．

本研究では，意味カテゴリ間の制約として，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 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，輸送用機器という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 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．

2 提案手法

ここでは，本研究がベースラインとして採用したブートストラッピングの既存手法として Carlson らによる CPL アルゴリズム [1] を説明し，続いて提案手法である意味カテゴリ間の階層関係を用いたフィルタリング手法を説明する．

2.1 CPL アルゴリズム

本研究で採用したブートストラッピング手法は，インスタンス候補の抽出，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．インスタンス候補の抽出では，コーパス中でシードインスタンスと共起するパターンや，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．なお，本研究ではシード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，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．Carlson らによれば，この制約は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て，一般的なパターンや曖昧性の高い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 1 で定義される適合率に基づき，ランキングする．

$$Precision(p) = \frac{\sum_{i \in c} count(i,p)}{count(p)} \quad (1)$$

ここで，c はある意味カテゴリのシード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，インスタンスは上位 100 位内に入ったものを，パターンは上位 10 位以内に入ったものを獲得する．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 つ以上のシードインスタンス・パターンと共起することとする．

2.2 上位カテゴリのパターン

提案手法では，これまでに説明したブートストラッピング手法でのフィルタリングに加え，さらに上位カテゴリのパターンを用いたフィルタリングを行う．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．例えば自動車の車種のカテゴリは自動二輪の車種などの属する，輸送用機器という上位カテゴリをつ．この輸送用機器カテゴリは「X に乗る」や「X の燃費」などのパターンを持っている．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する「インサイト」や「ヴィッツ」など輸送用機器の性質を持つもののみがインスタンス候補として残される．

この上位カテゴリのパターンは，上位カテゴリが共通であるカテゴリ群（兄弟カテゴリ）のシードインスタンスを用いて次のように決定する．まず兄弟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先に記したブートストラッピングで用いられ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 3 に示す．すなわち表 3 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兄弟カテゴリに共通のものにするため，2 つ以上のカテゴリで出現するものに限定する．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考える．細かい意味カテゴリの認定は，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，傘下のカテゴリに共通のパターンである必要があるので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．このことから兄弟カテゴリの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ため，以下の式 2 によりスコア付けを行う．式 2 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．

$$Score(C,p) = Entropy(C,p) * Recall(C,p) \quad (2)$$

$$Entropy(C,p) = -\sum_{c \subset C} P_c(p) \log_{|C|} P_c(p) \quad (3)$$

$$Recall(C,p) = \frac{\sum_{c \subset C} cooccur(p,c)}{\sum_{c \subset C} |I_{s_c}|} \quad (4)$$

$$P_c(p) = \frac{cooccur(p,c)}{\sum_{c \subset C} cooccur(p,c)} \quad (5)$$

$$cooccur(p,c) = |I_{s_c} \cap I_{p_c}| \quad (6)$$

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．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．本論文ではシードインスタンスのカバー率が 90 %を超えるまでパターンを獲得する．

3 実験

3.1 実験設定

実験にはコーパスとして，ウェブページ 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 36 個の意味カテゴリを表 1 の左端に記した．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 1 では上位カテゴリが共通のカテゴリ毎に分けて記している．シードインスタンスは各カテゴリで 15 個とした．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 [9] を Wikipedia に適用し，その出力結果のノイズを人手で除去した．

	獲得インスタンス数		適合率(%)	
カテゴリ	ベースライン	提案手法	ベースライン	提案手法
道路	24	24	37.5	37.5
橋	300	81	0.7	3.7
鉄道路線	46	37	91.3	91.9
日本の都市	5	5	100.0	100.0
アメリカ合衆国の都市	19	17	73.7	76.5
中華人民共和国の都市	10	9	10.0	11.1
アフリカの国	35	28	37.1	35.7
ヨーロッパの国	5	5	80.0	80.0
アジアの国	6	6	66.7	66.7
精神疾患	29	26	41.4	42.3
感染症	112	107	16.1	15.9
細菌	21	17	38.1	23.5
ウイルス	23	20	0.0	0.0
劇場	34	8	20.6	0.0
美術館	60	10	15.0	20.0
動物園	5	5	0.0	0.0
遊園地	16	15	62.5	66.7
水族館	0	0	0.0	0.0
公園	17	17	0.0	0.0
自動車メーカー	30	13	73.3	76.9
医薬品メーカー	0	0	0.0	0.0
自動二輪の車種	49	49	22.4	22.4
自動車の車種	200	157	47.0	51.0
神社	107	100	22.4	24.0
寺	0	0	0.0	0.0
ラジオ局	8	3	12.5	33.3
テレビ局	59	57	49.2	50.9
池	0	0	0.0	0.0
湖	63	61	23.8	23.0
河川	27	24	77.8	83.3
山地	11	8	54.5	37.5
島	84	83	19.0	19.3
空港	10	9	70.0	66.7
鉄道駅	15	14	46.7	42.9
元素	13	12	23.1	25.0
化合物	23	21	47.8	47.6
total	1466	1048	29.0	34.4

表 1: 提案手法とベースラインの各カテゴリにおける獲得インスタンスと適合率

Carlson らの手法をベースラインとし，提案手法と比較する．ブートストラッピングの反復を 5 回行い，システムが獲得した全インスタンスを人手により評価する．

3.2 実験結果

提案手法とベースラインでの各カテゴリにおける獲得インスタンスの数と適合率を表 1 に示す．総獲得インスタンス数に対するベースラインでの適合率は 29.0 %であるのに対し，提案手法での適合率は 34.4 %と向上した．表 2 には上位カテゴリを共有するカテゴリで獲得した共通のパターンの例を示し，表 3 にはアメリカ合衆国

カテゴリ	共通のパターン
アメリカ合衆国の都市，中華人民共和国の都市，日本の都市	行く，住む，ホテル，開催，街
精神疾患，感染症	治療，症状，診断，原因，病気

表 2: 上位カテゴリを共有するカテゴリにおける共通のパターン

カテゴリ	手法	インスタンス
アメリカ合衆国の都市	ベースライン	ロサンゼルス，ニューヨーク，サンアントニオ，シカゴ，サンディエゴ，ポートランド，バンクーバー，LA，ロス，ホノルル市，パームスプリングス，仙台市，ミュンヘン，NY，クリネックススタジアム宮城
	提案手法	ロサンゼルス，ニューヨーク，サンアントニオ，シカゴ，サンディエゴ，ポートランド，バンクーバー，LA，ロス，ホノルル市，パームスプリングス，仙台市，ミュンヘン，NY，フィラデルフィア
精神疾患	ベースライン	過労死，診断，SE，育児ノイローゼ，アルコール依存症，睡眠障害，躁うつ病，社会障害，パニック，PTSD，うつ状態，ADHD，躁鬱病，てんかん，公汎性発達障害
	提案手法	過労死，診断，過食症，育児ノイローゼ，アルコール依存症，睡眠障害，躁うつ病，社会障害，パニック，PTSD，うつ状態，ADHD，躁鬱病，てんかん，公汎性発達障害

表 3: 提案手法とベースラインにおける獲得インスタンス（上位 15 個）

の都市と精神疾患という 2 つのカテゴリについて，獲得された順に 15 個のインスタンスを示す．表 3 に示したとおり，ベースラインではアメリカ合衆国の都市において「クリネックススタジアム宮城」を獲得してしい，また精神疾患においては「SE」という，本来の意味カテゴリとは大きく異なるインスタンスを獲得してしまっている．これに対して提案手法では，都市や病気には属さないインスタンスを獲得していない．すなわち表 2 を用いたフィルタは上位カテゴリの性質を保有しないインスタンスを除去するフィルタとして働いていると考えられ，提案手法は有効であると言える．

しかしながら劇場のカテゴリにおいては正解インスタンスをすべてフィルタリングしてしまっている．これは複数の上位カテゴリで出現しているパターンは最も頻度の高いカテゴリの候補とするという制約を用いた結果，劇場と美術館での共通のパターン候補が 3 つしか得られず，正解インスタンスを網羅的にカバーできるパターンが獲得できなかったためであると考えられる．また，パターンが得られなかったことについては美術館と劇場の共通性が少なく，共通のパターンがもともとあまりなかった可能性もある．複数の上位カテゴリに出現するパターンがどのカテゴリに属するかや，カテゴリがどの上位カテゴリに属するかを決定する方法については，再検討する必要がある．

4 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，カテゴリ間の関係を用いる集合拡張法の最新の研究である 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．

しかし本論文で提案した上位カテゴリのパターンによるフィルタはブートストラッピングの前にシードインスタンスによって定められ，更新されることがない．このためインスタンスがまったく獲得できない事態や，十分なインスタンスが獲得できないまま収束してしまう可能性がある．これに対処するために，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利用することや，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を検討することが今後の課題である．

謝辞 本研究は，文部科学省科研費（23240018），文部科学省科研費（23700159），および JST 戦略的創造研究推進事業さきがけの一環として行われた．

参考文献

[1] 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.

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

[3] 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*, pp. 1011–1020, 2008.

[4] 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*, pp. 2–6, 2011.

[5] 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*, pp. 938–947, 2009.

[6] 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*, pp. 113–120, 2006.

[7] 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*, pp. 290–298, 2009.

[8] 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*, CIKM '09, pp. 225–234, 2009.

[9] 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, pp. 3–24, 2009-07-10.