ブートストラップを用いた検索クエリログからの意味カテゴリ獲得の分析

牧本慎平† 小町守‡ 颯々野学†

† ヤフー株式会社
{smakimot,msassano}@yahoo-corp.jp

‡ 奈良先端科学技術大学院大学 情報科学研究科
mamoru-k@is.naist.jp

1 はじめに

広告のマッチングやウェブ検索における意図の推定などで高い性能を得るためには，大規模な固有表現辞書や意味カテゴリの知識が必要不可欠である．近年，それらの知識を得る一手法として，**ブートストラップ**による，カテゴリ付きの固有表現を獲得が提案されている [3, 5]．意味カテゴリ獲得におけるブートストラップは，シードとして同一カテゴリに属す少数の固有表現などのインスタンスを設定し，インスタンスからのパターン抽出とパターンからのインスタンス抽出との反復によって，大規模なインスタンス集合を獲得する手法である．ブートストラップは，小規模なデータを与えることで，大規模なデータを獲得できるという利点がある．

我々は検索クエリログから意味カテゴリ付きのインスタンス獲得を対象とした．検索クエリは検索エンジン利用者が検索を行なう際に入力した文字列である．検索はインターネット利用者のほとんどが行なう活動であり，そのログは彼らの関心を直接的に表現している．そのため，新語やマイナーな固有表現などが多く含まれ，大規模な固有表現辞書を作成するために有用である．

ブートストラップによる意味カテゴリの獲得を行なう際の問題点として，対象とするカテゴリや使用するシードやパラメータによって性能が大きく異なるということがある．特に性能低下の原因となっているのは，**意味ドリフト** (semantic drift) と呼ばれる現象である．これは，反復を繰り返していくうちに，「**画像**」や「**東京**」のように複数のカテゴリで出現する一般的なパターン（**ジェネリック・パターン**）や複数のカテゴリに出現する曖昧なインスタンスを獲得してしまい，それによって以降の反復で獲得できるインスタンスのカテゴリが遷移してしまうというものである．意味ドリフトの発生を抑制するためにはパラメータのチューニングが必要不可欠であるが，獲得したインスタンスが正しいカテゴリに属しているかを評価するためには正解データを用意する必要があるため，性能評価にコストがかかる．

本稿では，反復を繰り返すことによって獲得される意味カテゴリがどのように遷移していくかの調査を行なった．従来，ブートストラップの研究では，目的のカテゴリに属すインスタンスをいかに集めるかに重点を置いていたが，我々は意味カテゴリがどのように遷移して行くかを分析することによって，意味ドリフトの発生を抑制する方法を考察する．本来，反復を繰り返すことで，目的のカテゴリから別のカテゴリに遷移する現象を意味ドリフトと呼ぶが，本稿では，目的としているカテゴリかどうかにかかわらず，獲得されたインスタンス集合の傾向が別のカテゴリに移ることをカテゴリ遷移と呼称する．

我々は各インスタンス集合に含まれるインスタンスそれぞれに関連付けられたキーワードの類似度合いを調査することによって，インスタンス集合間の類似度を測り，それによってカテゴリの遷移の傾向の分析を行なった．今回，インスタンスに関連付けられたキーワードとして日本語版 Wikipedia の各見出し語に付与されたカテゴリを用いた．

2 関連研究

固有表現・意味知識の収集を大規模コーパスから（半）自動で行なう研究は古くから行なわれてきた．

Hearst [1] は人手で構築したパターンをシードとしてブートストラップ的に知識獲得を行なう手法を初めて提案した．

検索クエリログを対象とした意味知識獲得に関しては，Sekine ら [6] の英語の検索クエリログを用いた，人手で作成された固有表現辞書の精査などがある．小町らの *Tchai*[3] は，汎用なブートストラッピングアルゴリズム *Espresso*[5] をもととした，検索クエリログからカテゴリ付きインスタンスを獲得するための手法である．また，クエリログとウェブ文書から固有表現のカテゴリとその属性を獲得する手法を Paşca らが提案している [4].

3 *Tchai* アルゴリズム [3]

我々は，獲得されるインスタンス集合の遷移の解析を行なう上で，検索クエリログを対象として高い性能を示している *Tchai* アルゴリズムを使用した．

3.1 アルゴリズムの概略

Tchai は特定のカテゴリに属す少量のシードインスタンスから開始する．そのインスタンスと共起するパターンを獲得し，信頼度を計算する．次に，パターンと共起するインスタンスを獲得し，パターンの信頼度から新たに獲得されたインスタンスの信頼度を計算する．そして，信頼度の高いインスタンスを目的のカテゴリに属すインスタンスとして獲得する．その後，獲得されたインスタンスの信頼度をもとに，パターンの信頼度を再計算する．このように，インスタンス抽出・信頼度計算とパターン抽出・信頼度計算を繰り返すことにより，大規模なカテゴリ付きのインスタンスを獲得する．

インスタンス集合 I 内のインスタンス i の信頼度 $r_\iota(i)$ の計算には以下の式を用いる．

$$r_\iota(i) = \frac{\sum_{p \in P} \frac{\mathrm{pmi}(i,p)}{\max_{p \in P} \mathrm{pmi}} r_\pi(p)}{|P|} \quad (1)$$

また，パターン集合 P 内のパターン p の信頼度 $r_\pi(p)$ の計算には以下の式を用いる．

$$r_\pi(p) = \frac{\sum_{i \in I} \frac{\mathrm{pmi}(i,p)}{\max_{i \in I} \mathrm{pmi}} r_\iota(i)}{|I|} \quad (2)$$

ここで使用されている $\mathrm{pmi}(i,p)$ は i と p の相互情報量 (pointwise mutual information, PMI) であり，下式で表わされる．

$$\mathrm{pmi} = \log \frac{|i,p|}{|i,\ ||\ ,p|} \tag{3}$$

3.2 *Tchai* の簡易化

本稿で行なった実験では，*Tchai* に対して，2 単語で構成されるクエリのみを対象として動作するように変更を加えた．これは実装の簡易化やノイズの除去を目的としている．

通常の *Tchai* の場合，例えば，インスタンス「みずほ銀行」からパターンを抽出する際，クエリ「**みずほ銀行 銀行番号**」からは「**# 銀行番号**」というパターンが取れる．ここで，# はインスタンスが入るスロットである．そのため，単純な方法では文字列処理が必要となるため計算量が膨大になり，処理効率に時間がかかる．

我々は，クエリを 2 単語に限定し，(**みずほ銀行**, **銀行番号**) のように 1 つのクエリを (T_l, T_r) という形式で表現することによって，実装を単純化することにした．これにより，事前のインデクス化を簡単に行なうことが可能になる．また，これによりインデクスを使用しない通常の *Tchai* の実装と比較して約 85% 実行時間を削減できるという付加効果を得た．

また，この手法はクエリログのノイズをある程度除去することに貢献する．2 つの単語で構成されたクエリは明確に分かち書きされた 2 つの独立した単語であることが多い．例えば，クエリ「**analysis**」に対し，インスタンス「**ana**」からパターン「**#lysis**」が取れてしまうという可能性を減らすことができる．また，クエリログ中に機械的に入力された意味を持たない文字列をフィルタリングすることができる．

4 インスタンス集合間の類似度の導入

ブートストラップによる知識獲得では停止条件として反復ごとに獲得されたインスタンスを評価する方法が取られてきた．例えば，*Espresso* では，パターン抽出時に計算する信頼度の平均値と前回の反復での値との割合が閾値以下であれば停止するという方法を取っている．我々の関心はブートストラップのカテゴリ遷移の生じる過程の分析であるので，停止条件の検討とは目的が異なる．そこで，意味的距離を測るためにインスタンスに結び付いた複数個のカテゴリを用いた類似度の測定方法を導入した．一般的に獲得される語は曖昧な場合が多いため，意味的類似度を測るために，語に結び付いたカテゴリの情報を類似度に使用した方が，単純な bag of words で行なうより有効な性能を示すと考えたからである．

今回，インスタンスと結び付いた情報として用いたのは Wikipedia のそれぞれの見出し語に付与されているカテゴリである．Wikipedia では各見出し語ごとにその語の属性を説明したカテゴリが複数個付与されている．例えば「**みずほ銀行**」という見出し語に対しては，「**都市銀行**」「**みずほフィナンシャルグループ**」「**東京都の銀行**」など 6 つのカテゴリが付与されている．そのため，インスタンス集合の類似性を測る上で，ある程度の曖昧さを考慮することが可能となる．

2 つのインスタンス集合間の類似度を測るためにコサイン類似度を用いた．コサイン類似度は比較したい 2 つのドキュメントの単語ベクターのコサイン距離を測ることにより類似度を得る手法であり，ドキュメントの単語

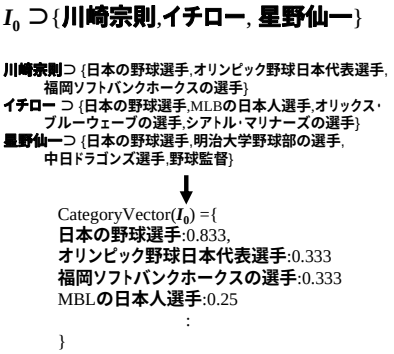


図 1 カテゴリベクターの作成の例

ベクター $\mathbf{d}_1$，$\mathbf{d}_2$ について，下式で表わされる．

$$\mathrm{sim}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1 \ \mathbf{d}_2}{|\mathbf{d}_1||\mathbf{d}_2|} \tag{4}$$

我々の提案する尺度は，単語ベクターとして各インスタンス集合のカテゴリ情報を表現している．まず，シードインスタンスの集合および各反復で獲得できるインスタンス集合のなかから，Wikipedia の項目名に合致するものを探す．それから，その項目に付与されているカテゴリを調べ，インスタンス集合全体で，どのカテゴリがいくつ含まれているかを数え上げことにより，インスタンス集合からカテゴリベクターが作成される．これらカテゴリベクター間のコサイン類似度を求めることによって，カテゴリ遷移の度合いを計測する．図 1 はカテゴリベクターの作り方を例示したものである．

コサイン類似度を尺度として利用する際，今回，以下の 2 つの尺度を用いて調査を行なった．

シード類似度 各反復でのインスタンス集合のカテゴリベクターとシードとなったインスタンス集合のカテゴリベクターとを比較する．

差分類似度 各反復でのインスタンス集合のカテゴリベクターとその直前の反復でのインスタンス集合のカテゴリベクターとを比較する．

いずれの尺度も，コサイン類似度をもとにしているため，0 以上 1 以下の値を取る．

シード類似度はシードからどれだけ意味カテゴリが遷移していったのかを計測する．この類似度を導入することで，正解インスタンスを用意することなく，シードとそのカテゴリを使って評価することができるという利点がある．

一方，**差分類似度**は直前の反復からどれだけ意味カテゴリが遷移していったかを測っている．極端に数値が下がった位置でカテゴリ遷移が起きていると考えられる．差分類似度で極端な数値の変化が生じた場合でも，カテゴリがシードに近いものに遷移する場合も考えられる．*Espresso* のような反復の停止条件として前回の反復との違いを用いる手法の場合，カテゴリの遷移によって獲得できるインスタンス集合が改善する場合を捉えることができない．そこで，差分類似度が極端な変化を生じた回のシード類似度を観察することで，その時点でのカテゴリ遷移によって収集したインスタンス集合がカテゴリに合致する方向に遷移したのか，合致しない方向で遷移したのかが分かるようになる．

表 1　使用したシードのリスト.

カテゴリ	シード
旅行	ana，his，jr，jtb，じゃらん
自動車メーカー	bmw，トヨタ，ダイハツ，ホンダ，マツダ
女性声優	茅原実里，平野綾，田村ゆかり，水樹奈々，坂本真綾
野球選手	川崎宗則, イチロー, 星野仙一, ダルビッシュ有，山本昌

5　実験

5.1　データセット

■検索クエリログ　Yahoo! 検索[*1] の 2008 年 8 月分の検索クエリログのうち，空白文字で区切られた 2 つの単語で構成されたものを用いた．我々が解析の対象とした検索クエリログ 1 ヶ月分のうち，2 単語で構成されたクエリは全体の約 40% であり，十分な量がサンプリングできると判断できる．本稿では，2 単語クエリの頻度上位 1,000,000 異なりクエリのデータを対象とする．検索クエリには全角英数字を半角にするなどの正規化を行なった．

■カテゴリ付き辞書　カテゴリ付きの辞書として日本語版 Wikipedia のダンプ（2008 年 7 月 24 日版）[*2]を使用した．これは，760,764 項目の記事があり，51,539 種類のカテゴリを持っている．また，Wikipedia の見出し語がクエリログデータ中に含まれる単語をカバーしている率は 31.31% である．一方，クエリログデータ内の単語が Wikipedia の見出し語をカバー率している率は 11.99% である．

5.2　実験設定

3.2 節で述べた簡易版の *Tchai* の実装を使用する．

反復で出力されたインスタンスの集合から毎回，信頼度順で上位 500 個を分析の対象とする．反復回数は 50 回とする．また，全てのパターンを用いてインスタンスの信頼度計算を行なった[*3]．

「旅行」「自動車メーカー」「女性声優」「野球選手」の 5 種類のカテゴリで解析を行なった．「旅行」は小町ら *Tchai*[3] との比較のために対象とした．他の 3 カテゴリについては各カテゴリに属している 1 単語クエリの上位 5 件を採用した．それぞれのシードインスタンスは表 1 に示す．「自動車メーカー」「女性声優」「野球選手」は Wikipedia に対応するカテゴリのリストが存在するため，Wikipedia のカテゴリに含まれるインスタンスを正解と考えて精度を計算する．今回，獲得上限数が決定されているので，再現率は求めない．

5.3　結果

以上の設定で調査したカテゴリ毎の類似度の変化を図 2 で示した．「旅行」カテゴリ（図 2(a)）以外のグラフでは Wikipedia のカテゴリをもとにした精度を示した．精度は正解データとした Wikipedia の網羅性が低く，10% 未満である事例が多くあったので，「自動車メーカー」「女性声優」に関しては値を 10 倍，「野球選手」に関しては 100 倍にして図にプロットした．

[*1] http://search.yahoo.co.jp/

[*2] http://download.wikimedia.org/jawiki/20080724/jawiki-20080724-pages-articles.xml.bz2

[*3] Komachi ら [2] によると，全てのパターンを用いても，信頼度上位のもののみを用いる場合と本質的に同じである．そこで，カテゴリ遷移の性質を明確に表現するためこの設定を用いた．

6　議論

以上の実験を行なった上での議論を述べる．

Wikipedia の対応するカテゴリから精度を調査した「自動車メーカー」（図 2(b)），「女性声優」（図 22(c)），「野球選手」（図 2(d)）を見ると，シード類似度とその反復で取得されたインスタンスの精度に，相関があることが示唆される．

一般に精度を計算するにはラベル付きデータが必要であるが，シード類似度を測定するにはシードのみの情報があれば良いため，性能を評価するコストが少なくて済む利点がある．

差分類似度はその前の反復からの違いを表現しており，大きな数値の下降が見られる位置において，大規模なカテゴリ遷移が生じていると考えられる．

例えば，「野球選手」カテゴリ (図 2(d)) において，5 回目の反復ステップを見ると差分類似度が大きく下降している．これは大規模なインスタンスの入れ替えが生じているためである．ここで，シード類似度も同様に大きく下降していることから，新たに獲得されたインスタンスに目的カテゴリ以外のものが多く含まれていることが予想される．

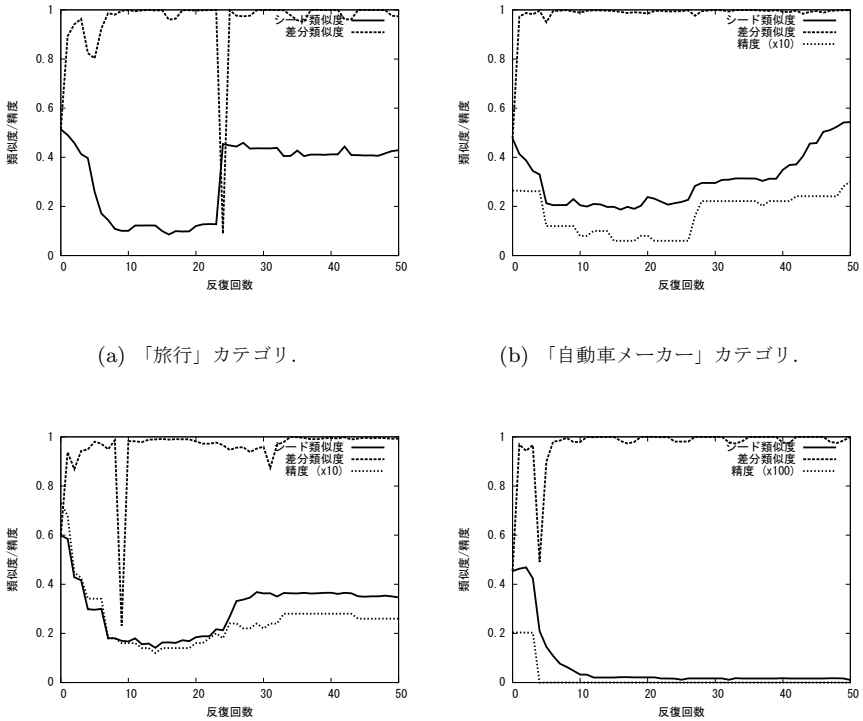
そこで，5 回目の反復において入れ替わった 236 個のインスタンスの精査を行なった．その結果，新たに獲得されたインスタンスの中に野球選手は含まれていなかった．一方，この反復で消えたインスタンスの中には「王貞治」「松井秀喜」など野球選手や野球関係者が 19 個，「ジャイアンツ球場」「星野ジャパン」など野球関連の用語 32 個があった．これらのことから，この反復ステップで目的のカテゴリに近い内容がインスタンスの上位 500 件から消えたことになり，信頼度上位のインスタンスの傾向が変化したことが言える．従って，差分類似度とシード類似度がともに大きく下降したこの反復ステップで，大規模なカテゴリ遷移が生じていることが確認された．

これら 5 つのカテゴリで共通して見られる特徴として，ブートストラップの反復初期に，差分類似度の大きな下降が生じていることがある．これは，ブートストラップ初期の段階でカテゴリの遷移が生じることを示している．

しかしながら，差分類似度において大きな下降が生じた部分で必ずしもシードとの類似度が悪化しない場合がある．例えば，図 2(a) では差分類似度が大きく低下した反復回において，シード類似度の大幅な上昇が起きている．これらの現象が起きるのは，求めるカテゴリに属しているもの以外のインスタンスが遷移してる，あるいは，カテゴリ内で遷移しているなどの理由が考えられる．

次に，ブートストラップにおいて，反復を繰り返していく過程でのカテゴリ遷移の生じ方を解析した．「旅行」「女性声優」「野球選手」カテゴリにおいては，今回観測した 50 回の反復の後半において差分類似度・シード類似度双方がある程度同じ値で落ち着いている．一方，「自動車メーカー」カテゴリにおいては，反復回数 50 回の時点でシード類似度が上昇傾向にある．これは，「自動車メーカー」カテゴリが 50 回の反復内ではまだ安定していないことが原因だと言える．また，差分類似度を観察すると，「旅行」「野球選手」のカテゴリにおいては，一定の頻度で低下して復帰するという揺らぎを示しており，カテゴリの入れ替えが定期的に生じていることが考えられる．

Komachi ら [2] は *Espresso* の反復を繰り返すことによって，獲得されるパターン・インスタンスは定常状態



(a) 「旅行」カテゴリ.

(b) 「自動車メーカー」カテゴリ.

(c) 「女性声優」カテゴリ.

(d) 「野球選手」カテゴリ.

図 2　カテゴリ毎の類似度の変化

になることを指摘しているが，我々も *Espresso* ベースの *Tchai* を使用しているので，同様に反復を繰り返すことによって定常状態になると考えられる．最も高い精度はいずれのカテゴリにおいても反復初期に得られている．一般的に，意味ドリフトが生じてカテゴリが遷移するたびに性能が低下して行くので，その前に反復を停止するのが重要である[*4]．

7　まとめ

本稿では，ブートストラッピングアルゴリズムで獲得された意味カテゴリのカテゴリ遷移の度合いを測る尺度を導入した．そして，複数のカテゴリを対象として，ブートストラップの反復ごとに導入した尺度を用いてカテゴリ遷移を分析した．

シード類似度は獲得されたインスタンス集合がシードとどれくらい類似しているかを測る尺度であり，本稿では実際の精度との相関を示した．これを用いることによって正解データを持たない場合でも，インスタンス集合とシードとの意味的な類似度を推定可能になる．

また，差分類似度は直前の反復で獲得されたインスタンス集合との差異であり，これを用いることで，カテゴリ遷移の傾向を調査した．その中で，反復を繰り返すうちに，取得されるインスタンス集合がある定常状態になるということを確認した．

今回，インスタンスに関連付けられたキーワードとして Wikipedia のカテゴリを使用したが，この手法では対応できる用語が限られているため，対象によっては正しい評価ができない場合が考えられる．例えば，製品の型番などは Wikipedia の見出し語として個々に登録されておらず，そのようなカテゴリを求める場合，Wikipedia のカテゴリを使用するのは不向きである．今後は，検索結果のスニペットなどから得たキーワードを用いて，より正確な類似度判定を行ない，自動的に最適なインスタンス集合を発見する方法を検討していきたい．

[*4] 図 2(a) のようにある段階でシード類似度が上昇する事例もあるが，その現象を制御する方法がないため一度性能が低下した時点で停止するのが現実的である．

参考文献

[1] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pp. 539–545, 1992.

[2] M. Komachi, T. Kudo, M. Shimbo, and Y. Matsumoto. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proc. of EMNLP*, pp. 1011–1020, 2008.

[3] 小町, 鈴木. 検索ログからの半教師あり意味知識獲得の改善. 人工知能学会論文誌, 23(3):217 – 225, 2008.

[4] M. Paşca and B. Van Durme. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proc. of ACL-08: HLT*, pp. 19–27, 2008.

[5] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of COLING/ACL-06*, pp. 113–120, 2006.

[6] S. Sekine and H. Suzuki. Acquiring ontological knowledge from query logs. In *Proc. of WWW*, pp. 1223–1224, 2007.