意味的類似度を用いたWeb文書からの集合拡張

萩原 正人　関根 聡
楽天技術研究所
{masato.hagiwara, satoshi.b.sekine}@mail.rakuten.co.jp

1 はじめに

少数の正解セットをシードとして与えて，それらと同一の意味的語彙カテゴリ (以下単に意味カテゴリと呼ぶ) に属する語の集合を獲得するタスクは「集合拡張」と呼ばれる．例えば“日本”, “米国” などの正解セットから，“韓国”, “中国”, “ロシア”, “イギリス”, ... など国・地域名の集合を獲得する．対象とする集合の要素を以下では「インスタンス」と呼ぶ．集合拡張には幅広い応用があり，固有表現獲得や辞書構築等に用いられてきた．これまで，Google Sets[1], SEAL (Set Expander for Any Language)[5] などの各種手法が提案されているが，主にWebページ等の半構造化文書の構造に基づいて類似関係を獲得しているため[2]，“その他”, “もっと見る” など，Webページのナビゲーション等のリスト構造に共起しやすいが，意味的にはシードと無関係な語が獲得されてしまうという問題がある．また，この集合拡張のベイズ的アプローチであるBayesian Sets [1] も提案されているが，最初から閉じたインスタンス集合と素性の共起を仮定しており，シードから新たにインスタンスを獲得する目的には適さない．

一方，少数のシードから，文脈パターンを手がかりとしたブートストラップ法を用いて，半教師あり学習により固有表現やその関係を抽出する手法が注目されている．ブートストラップ法による意味カテゴリの抽出は，手法・タスク共に集合拡張と共通するところが大きい．これまでに，is-a関係など語の意味的二項関係を抽出する *Espresso* アルゴリズム [4]，日本語のクエリログから意味カテゴリを抽出する *Tchai* アルゴリズム [3]，グラフカーネルを用いてブートストラップを定式化した手法（以下 *g-Espresso* アルゴリズムと呼ぶ）[2] など，様々な手法が提案されている．萩原ら [7] は，グラフカーネルに基づく意味カテゴリ抽出法 *g-Monaka* アルゴリズム を提案した．*g-Monaka* アルゴリズムは，文字 *n* グラムを用いているため，形態素解析などを経ることなく，非分かち書き文から直接意味カテゴリを抽出できる．さらに,「両側隣接制約」という，右側と左側の両方の文脈を考慮した類似度を考慮することにより，さらに高い精度で意味的類似度を求めることができるという特長がある．

しかしながら，この *g-Monaka* アルゴリズムをWeb文書からの集合拡張にそのまま適用した場合，パターンとして用いている文字 *n* グラムのみでは識別力が低く，関連のあるインスタンスを効率よく見つけることができない．例えば，調理器具に関連する意味カテゴリを取得するために “中華鍋”, “圧力鍋” というシードを与えると，これらを特徴づける “#を買う”, “人気の#” などのパターンがWebから獲得される[3]．しかし，次にこれらのパターンと共起するインスタンスをWeb検索により再度獲得すると，“携帯”, “車” などの，文脈とは共起するがシードとは無関係なインスタンスが大量に獲得されてしまい，計算対象が膨大になる．したがって，何らかの方法で獲得対象のインスタンスを絞り込んでから，文脈パターンに基づくブートストラップ法を適用することにより，効率よく高精度なインスタンスを求めることができると考えられる．

これに対して本稿では，これら2つのアプローチを統合し，(1) 候補獲得 (2) 候補ランキングの2つのフェーズを用いた集合拡張を提案する．フェーズ (1) では，従来手法と同様に，Webページにおけるリスト構造等を用いて拡張候補を列挙する．フェーズ (2) では，獲得された候補とシード間の意味的類似度を上述の *g-Monaka* アルゴリズムを用いて計算し，候補インスタンスを再ランキングする．3種類のシードセットを用いた評価実験の結果，各種従来手法と比較して，高い精度・再現率で集合拡張できることが分かった．

2 SEAL アルゴリズム

SEAL アルゴリズムは，HTML や XML などの半構造化文書集合から集合拡張する完全に言語非依存なアルゴリズムであり，(1) 抽出 (2) ランキングの2つのフェーズから構成される．フェーズ (1) では，与えたシードを含む単一のWebページに注目し，各シードの左右の文字列から，ラッパー (wrapper) と呼ばれるページ固有の抽出ルールをまず構築する．例えば，“ford”, “nissan”, “toyota” の3つのシードを与えたとき，あるWebページのHTML中に

```
...<li class="ford"><a href="http:// ...
...<li class="nissan"><a href="http:// ...
...<li class="toyota"><a href="http:// ...
```

[1] http://labs.google.com/sets

[2] Google Sets についてはそのアルゴリズムは非公開ではあるが，結果の傾向から SEAL と同様のアルゴリズムを用いて抽出していると思われる．

[3] “#” はインスタンスが入るスロットを表す．“#を買う” のようなインスタンスの右側の文字列から成る文脈を右側文脈，“人気の#” のように左側のものを左側文脈と呼ぶ．

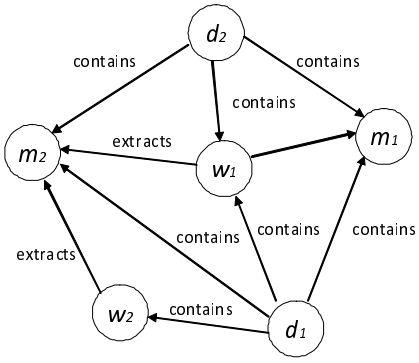

図 1: SEAL において構成されるグラフ

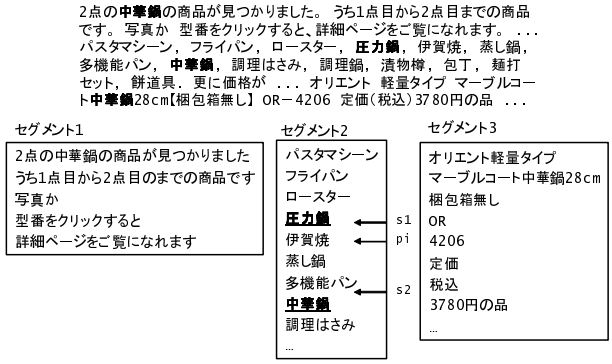

図 2: スニペットからの候補獲得

という文字列があった場合，各シードの左右の最長共通文字列，すなわち`<li class="[...]"><a href="http://`　をラッパーとして抽出する．次に，同一ページ内において，このラッパーのスロットに当てはまる文字列を全て取り出すと，

```
...<li class="honda"><a href="http:// ...
```

など他の箇所から “honda”, “acura” などの関連する他のインスタンスが獲得できる．

フェーズ (2) では，図 1 に示すような，全てのシード，全てのラッパー，全ての文書を節点とし，それらの間の抽出等の関係をエッジに対応させたグラフを構築する．あるラッパーからより多くの信頼度の高いインスタンスが抽出されれば，そのラッパーの信頼度も高い（その逆も真）という原則に従い，減衰付き PageRank に似たグラフ上のランダムウォークによる類似度伝播により，インスタンスを再ランキングする．3 言語, 36 ベンチマークを用いた評価実験では，Google Sets の 2 倍の平均精度を示すことが報告されている．

3　Web 文書からの集合拡張

前節で述べた SEAL によって抽出されるラッパーは，HTML タグなど，主に文書構造に依るものであるため，1 節で述べたように “その他”, “もっと見る” のような言語的に関係しているとは言い難いインスタンスを獲得してしまい，ラッパーからインスタンスを抽出しているという関係からグラフが構築される以上，フェーズ (2) の再ランキングによってこれらのインスタンスは高いスコアを与えられてしまう．

この問題は，半構造化文書の文字列をそのまま用いるのではなく，それぞれのインスタンスに対する言語現象としての文脈に注目し，意味類似度によって再ランキングすることにより解決できると考えられる．本節の提案手法では，フェーズ (1) においては SEAL をより簡単化した方法で候補を獲得し，フェーズ (2) において，*g-Monaka* アルゴリズムと同様のグラフカーネルを用いて再ランキングすることにより，高精度な集合拡張を実現している．

3.1　候補獲得

集合拡張候補は，以下のようにして抽出する．まず，検索エンジンに対して，全てのシードをスペース区切りで連結させたものをクエリとして入力し，検索結果の上位 300 件のスニペットのリストを得る．検索結果としては，シードを全て含んだ Web ページが得られる傾向があり，図 2 のように，与えたシードを含むようなリストを含むものが多く見られる．したがって，同一のリストに含まれる他の要素を抽出することにより，SEAL のような手法を用いなくても簡単に候補インスタンスを抽出できる．次のフェーズと組み合わせることにより抽出候補を再ランキングするため，本フェーズでの抽出精度はあまり重要ではない．

次に，得られたスニペットを Unicode の NFKC により正規化，小文字に統一し，区切り “...” によって複数の文字列セグメントに分割した．検索結果上位 300 件において重複している文字列セグメントは除外した．さらに，各文字列セグメントを，記号および句読点によって分割し，文字列リストとする．その結果が図 2 のセグメント 1, 2, 3 である．このうち，セグメント 1 と 3 のように，通常の文から生成されたようなセグメントは候補インスタンス抽出に不適であるため除外する．このようなセグメントの特徴として，各要素の長さが揃っていないという特徴があるため，要素の文字列長（Unicode 文字数）の標準偏差が 5.0 以上のものは除外した．

最後に，セグメントに含まれるインスタンス候補 i のスコア s_i を $s_i = \max_j \exp(-\alpha|p_i - s_j|)$ とした．ここで，p_i はインスタンス候補 i の位置，s_j は j 番目のシードの位置である．これはすなわち，最も近いシードとの距離に従い指数的に減衰するスコアを各インスタンス候補に与えるという意味である．減衰係数 α は $\alpha = 0.8$ とした．

3.2　候補ランキング

フェーズ (2) では，フェーズ (1) で獲得された候補を，*g-Monaka* アルゴリズムの類似度計算法により計算された意味的類似度により再ランキングする．

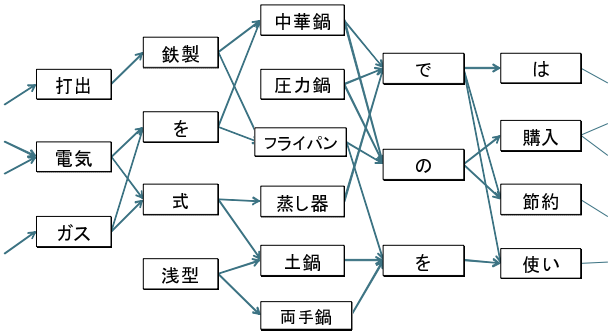

図 3: *g-Monaka* における n グラムの接続グラフ

まず，候補インスタンスのそれぞれについて，それらを検索クエリとして検索エンジンに与え，検索結果上位 300 件のスニペットから得られる文脈を抽出する．スニペットはフェーズ (1) と同様に NFKC により正規化・小文字化し，重複を取り除いた．また，日本語の割合が極端に少ない，記号が多すぎる，などの質の低いスニペットを除外するために，Google Web 日本語 N グラム第 1 版 [6][4]と同様の，文字種の割合によるフィルタリングを施した．

続いて，得られたスニペットの集合に含まれる全ての文字 n グラムについて，*g-Monaka* に従い，以下のように接続行列を構築する:

$$M(u,v) = \frac{\mathrm{pmi}(u,v)}{\max \mathrm{pmi}}, \quad \mathrm{pmi}(u,v) = \log \frac{|u,v|}{|u,*||*,v|} \tag{1}$$

ここで，$|u,v|$ は，n グラム u の後に n グラム v が続く頻度，$|u,*|$, $|*,v|$ はそれぞれ u,v そのものの出現頻度であり，ここではそれら自体を検索クエリとしたときの検索結果数の自然対数を取ったものを用いている．

次に，図 3 のように，全ての n グラムの集合 V を節点集合，M を接続行列として表現される有向重み付きグラフ G_M を考える．このグラフ上において，n グラム u と v に対して，右側文脈および左側文脈が類似しているほど，それらの意味は類似していると考えられる．右側文脈と左側文脈に基づく類似度行列は，接続行列 M を用いて，それぞれ

$$A_R = \frac{1}{|V|^2} MM^T, \quad A_L = \frac{1}{|V|^2} M^T M \tag{2}$$

として求められる．ここで，n グラム u と v に対して，右側文脈と左側文脈の両者が類似していてはじめて u と v は類似しているという「挟みこみ」の制約（以下，両側近接制約と呼ぶ）を加えるため，

$$A(i,j) = \sqrt[m]{\frac{1}{2}\left(A_R(i,j)^m + A_L(i,j)^m\right)} \tag{3}$$

として，要素毎の重み付き一般化平均によって A を求める．m は，この制約の強さを調節するパラメータであり，*g-Monaka* アルゴリズムにおいては $m = 0.1$ を用いた．最後に，この類似度行列 A を用いて，正則化ラプラシアンカーネル $R_\beta(A)$[2] を，

$$R_\beta(A) = \sum_{n=0}^{\infty} \beta^n (-\tilde{L}) \tag{4}$$

$$\tilde{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad D(i,i) = \sum_j A(i,j) \tag{5}$$

として求める．$R_\beta(A)$ の (i,j) 要素が，n グラム i と j の類似度に対応するため，シードベクトル $\boldsymbol{v}_0$(シードに対応する要素が 1, それ以外が 0 となっているようなベクトル) を用いて，$R_\beta(A)\boldsymbol{v}_0$ として最終的な類似度を計算する．

4 実験

4.1 評価手法

評価実験においては，人手により与えたいくつかのシードセットを用い，各手法の集合拡張結果を比較する．シードとしては，鍋 (NABE): {“中華鍋”, “圧力鍋”}, 映画ジャンル (MVGENRE): {“‘アクション”, “SF”, “ロマンス”}, 家電メーカー (EAMAKE): {“ソニー”, “パナソニック”, “東芝”}, の 3 セットを用いた．

フェーズ (2) の再ランキングにおいて，*g-Monaka* アルゴリズムの代わりに，分布類似度 (DS) や *g-Espresso* アルゴリズムを用いた場合を比較した．また，分布類似度は，両側近接制約の無い場合，すなわち，3.2 節の右側文脈による類似度 A_R と左側文脈による類似度 A_L を線形に結合したモデル ($m = 1.0$) と，両側近接制約のある場合 ($m = 0.1$) を比較した．最終的に比較した手法は，Google Sets, SEAL, DS($m = 1.0$), DS($m = 0.1$), *g-Espresso*, *g-Monaka*(提案手法) の 6 手法である．

評価は，情報検索の結果に対する評価指標である DCG (Discounted cumulative gain) を用いた．DCG は，検索要求に対する各結果の適合度の重み付き和であり，ランク i の適合度を r_i として，$DCG = r_1 + \sum_{i=2}^{50} r_i / \log_2 i$ として計算される．適合度 r_i は，同じ意味クラスに属する ($r_i = 2$)，同じ意味クラスには属さないが，文脈によっては相互交換可能である ($r_i = 1$), 相関が無い ($r_i = 0$) のいずれかとし，人手による評価を実施した．計算にはランク上位 50 個を使用した．

なお，Web 検索 API として，Yahoo! Japan の「ウェブ検索」API [5] を用いた．また，正則化ラプラシアンカーネルのパラメータは $\beta = 1.0 \times 10^{-2}$ に固定したが，このパラメータの値に対してカーネルによる類似度は非常に頑健であることが知られている [7]．文脈としては，文字 n グラム ($1 \le n \le 8$) を用いた．

4.2 結果

図 4 に，3 つのシードセットに対して各手法を用いた場合の DCG の値を示した．提案手法である *g-Monaka* が，平均して最も良い性能を示していることが分か

[4]フィルタリングの詳細については付属の README http://www.gsk.or.jp/catalog/GSK2007-C/GSK2007C_README.utf8.txt を参照

[5]http://developer.yahoo.co.jp/webapi/search/websearch/v2/websearch.html

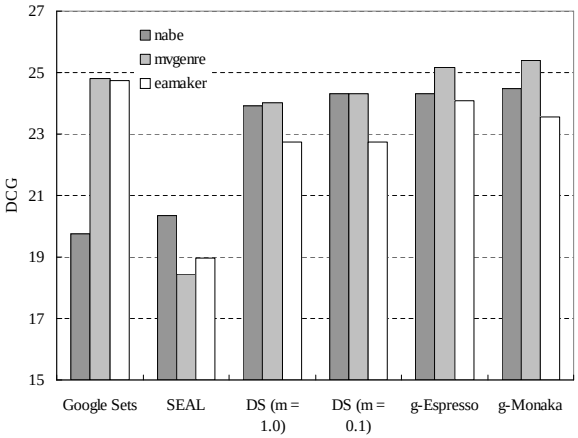

図 4: 各手法の精度比較

る．また，DS($m = 1.0$) と DS($m = 0.1$)，*g-Espresso* と *g-Monaka* をそれぞれ比較すると，前者（両側近接制約を用いないもの）よりも後者（両側近接制約を用いるもの）の方が平均して性能が高く，文脈による挟みこみがここでも有効であることが分かる．

表 1 に，Google Sets, SEAL, および提案手法の候補インスタンス (Phase1) および再ランキング後 (Phase2) に獲得されたインスタンスの上位 30 個を示した．Google Sets では，“その他” や “ブランド別” など，Web ページのリスト構造においてはシードと共起しやすいが，意味的には関連の無いものが数多く含まれている．また，SEAL の結果には，“ちゃんこ鍋” や “寄せ鍋” など，調理道具ではなく料理の種類として見た場合の鍋の名前が大量に含まれているが，シードには料理の種類としての意味は無く，これはシードとの意味的類似度を考慮していないためだと思われる．提案手法 (Phase1)，すなわちフェーズ (1) の候補抽出後の段階では，SEAL と同様にリスト構造を用いているため，“その他” などの無関係な語や，“調理器具” のような部分的にしか置き換え可能ではない語（この場合は上位語）が混在している．提案手法 (Phase2) は，*g-Monaka* によって再ランキングした後であり，上に挙げたような無関係な語がほとんど見られないことが分かる．

唯一，EAMAKER のシードセットで，Google Sets に比べて *g-Monaka* の精度が低下しているが，これは *g-Monaka* では “viera” “regza” “vaio” などの，各メーカーの製品ブランド名が獲得され，それらが適合度を低下させているのが原因である．メーカー名とブランド名は，例えば「ソニーを買う」などの文脈において相互交換可能であることが多いため，意味的類似度を用いた再ランキングにおいても完全に排除ができない．この傾向は MVGENRE のシードセットにおいても見られ，提案手法においても，例えば “12 モンキーズ” や “デジャヴ” などの具体的な映画名が混在する．

5 おわりに

本稿では，Web 文書からの集合拡張手法として，Web 文書のリスト構造に基づくアプローチと，グラフカーネルを用いて計算した意味的類似度に基づくアプローチを統合した手法を提案した．評価実験の結果，従来の集合拡張手法に比べ，意味的に類似している語が獲得でき，獲得精度が向上した．

なお，抽出されたインスタンスには，“ソニー”, “sony” などの表記ゆれや同義語が多く含まれ，これらの意味的類似度を考慮することにより獲得精度をさらに改善することができると考えられ，引き続き検討する．また，日本語以外の言語における性能評価は今後の課題である．

表 1: 獲得されたインスタンス (上位 30 個)

手法	インスタンス
Google Sets	圧力鍋, 中華鍋, 片手鍋, 両手鍋, フライパン, *その他, 保温鍋, 鍋・フライパンセット, 中華なべ, パスタ鍋, 玉子焼き器, *ih 調理器対応, ?ふた, *ブランド別, ?付属品・アクセサリ, フォンデュ鍋, *サイズ別, *素材別, *料理研究家・デザイナー別, *全商品, 土鍋, 鉄鍋, しゃぶしゃぶ鍋, 雪平鍋, 鍋・フライパン, 蒸し器, 寸胴鍋, *前に戻る, ジンギスカン鍋, *▼携帯からも！
SEAL	片手鍋, 土鍋, 行平鍋, ?慈善鍋, ?柳川鍋, ?社会鍋, 煎鍋, ?肉鍋, ?ちゃんこ鍋, ?寄せ鍋, 揚げ鍋, 蒸し鍋, もつ鍋, ジンギスカン鍋, 無水鍋, ?牡丹鍋, 大鍋, 手鍋, ?牛鍋, 平鍋, ?割れ鍋, すき焼き鍋, ?破鍋, ?寄鍋, 蒸鍋, 揚鍋, 鉄鍋, 寸胴鍋, 御鍋, せいろ
提案手法 Phase 1	中華鍋, 圧力鍋, フライパン, 雪平鍋, 寸胴鍋, 片手鍋, 土鍋, 蒸し器, パスタ鍋, 両手鍋, *鍋物用, 鍋, 鉄鍋, ケトル, *その他, 中華なべ, ?包丁, 揚げ鍋, 保温鍋, ?調理器具, 釜, 圧力なべ, 北京鍋, 親子鍋, しゃぶしゃぶ鍋, ミルクパン, 料理鍋, ティファール, パスタポット, *ホーム
提案手法 Phase 2	圧力鍋, 中華鍋, フライパン, 土鍋, 中華なべ, やかん, 雪平鍋, 蒸し器, 片手鍋, 圧力なべ, 北京鍋, 親子鍋, 餃子鍋, 天ぷら鍋, 鉄鍋, せいろ, 両手鍋, 寸胴鍋, パスタ鍋, ミルクパン, 鍋, ステンレス鍋, すき焼き鍋, ケトル, ?包丁, キャセロール, 銅鍋, 揚げ鍋, パスタパン

“?” は適合度 $r_i = 1$, “*” は $r_i = 0$ と判断されたインスタンスを表す

参考文献

[1] Zoubin Ghahramani and Katherine A.Heller. Bayesian sets. In *NIPS*, 2005.

[2] Mamoru Komachi, Taku Kudo, Masahi Shimbo, and Yuji Matsumoto. Graph-based analysis of semantic drift in *espresso*-like bootstrapping algorithms. In *Proc. of the EMNLP 2008*, pp. 1011–1020, 2008.

[3] Mamoru Komachi and Hisami Suzuki. Minimally supervised learning of semantic knowledge from query logs. In *Proc. of IJCNLP 2008*, pp. 358–365, 2008.

[4] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of COLING/ACL 2006*, pp. 113–120, 2006.

[5] Richard C. Wang and William W. Cohen. Language-independent set expansion of named entities using the web. In *Proc. of ICDM 2007*, pp. 342–350, 2007.

[6] 工藤拓, 賀沢秀人. Web 日本語 n グラム第 1 版. 言語資源協会, 2007.

[7] 萩原正人, 小川泰弘, 外山勝彦. グラフカーネルに基づく非分かち書き文からの意味的語彙カテゴリの抽出. 言語処理学会第 15 回年次大会講演論文集, pp. 697–700, 2009.