WWW から獲得した語の比喩的素描表現の上位下位関係精緻化に関する一考察

岩城 秀則 † 桝井 文人 ‡

† 北見工業大学大学院 工学研究科情報システム工学専攻

‡ 北見工業大学 工学部 情報システム工学科

†iwaki@ialab.cs.kitami-it.ac.jp ‡f-masui@mail.kitami-it.ac.jp

1　はじめに

新しい語やうろ覚えの語について調べたい場合，その正確な表記や, 意味, 特徴などを知りたいはずである．近年ではそれらを調べる際，Wikipedia や辞書サイトを概する WWW を利用することができるようになった．しかしながら，ユーザが求める情報までの経路が最適化されていないため，常に変化しつづける膨大な情報を持つ WWW からユーザの情報要求を引き出すことは難しい．

上記のような語彙把握の問題に対して我々[1, 2] は，WWW からクエリを表現できる語（デスクリプタ）を抽出し，クエリをデスクリプタの集合で比喩的に素描するシステム「Murasaki」を提案・実装している．クエリをデスクリプタ集合で表現することにより，ユーザが語の意味を把握する際，語の連想的理解が可能となる．

Murasaki にはデスクリプタを分類する機能が実装されており，抽出されたデスクリプタは「上位語」「属性語」「連想語」の 3 カテゴリに分類される．デスクリプタを分類することによって，ユーザが「語の意味」を把握する支援や，語の類似度計算支援 [3] なども期待できるが，そのためにはより高い分類精度の確保が不可欠である．

そこで本研究では，Murasaki のデスクリプタ分類性能の向上の可能性について議論する．具体的には，隅田ら [4] が提案している，Wikipedia から大量の上位下位関係にある語の組（以下，上位下位データと呼ぶ）を抽出する手法[1]の Murasaki への適用を考える．

予備調査を行ったところ，上位下位データと Murasaki が実際に抽出したデスクリプタの整合性が十分でないことが分かった．原因として，Murasaki と上位下位データが対象とする語の階層に差が生じている可能性が挙げられる．具体的には，Murasaki はクエリ「ナス」の上位語としてデスクリプタ「野菜」を得る．しかしながら，上位下位データには「ナス」の上位語として「ナス属」がエントリされている．どちらも正しい回答であるが，Murasaki に対して上位下位データはより詳細な概念であるといえる．

この階層の差を解消する方法として，Stoica ら [5, 6] が提案している *Castanet Algorithm* を用いる．*Castanet Algorithm* は，WordNet の上位下位階層の必要な部分のみを取り出すことによって，階層構造を洗練する手法である．この手法を上位下位データに用いることで，Murasaki 対象語との整合性の確保が可能となる．

以下，2 章で Murasaki の概要について説明し，3 章で事前に行った調査の内容と結果について述べる．更に 4 章で *Castanet Algorithm* の説明と，上位下位データへの適用の可能性について議論する．

2　Murasaki の概要

Murasaki は桝井ら [1, 2] によって提案・実装された，逐次型デスクリプタ抽出システムである．Murasaki はユーザが知りたい語を質問要求 (クエリ) としてシステムに提示することにより，システムが WWW 上からクエリを表現できる語 (デスクリプタ) を逐次的に抽出し，提示することで語をデスクリプタ集合で表現するシステムである．

Murasaki は,「のような」「という」「などの」といった統語パターンを利用して，WWW 上からデスクリプタを抽出し，デスクリプタ集合を構築する．ユーザが提示したクエリを統語パターンから「クエリ＋統語パターン」という WWW 検索表現を生成して，WWW 検索を行う．WWW 検索によって得られた Web ペー

[1]隅田らの手法は，2010 年 10 月 1 日より上位下位関係抽出ツール v1.0 として実装されており，本研究では上位下位関係抽出ツールを用いて隅田らの手法を実現した．

ジの snippet 中から「クエリ＋統語パターン＋名詞概念」という表現を認識する．名詞概念に相当する語句を抽出し，デスクリプタ集合を構築する．

Murasaki の分類は統語パターンを利用して実現している．図 1 に現在の分類アルゴリズムを示す．(Q はクエリ，d は任意のデスクリプタ，α，β，γ は検索件数における閾値とする)

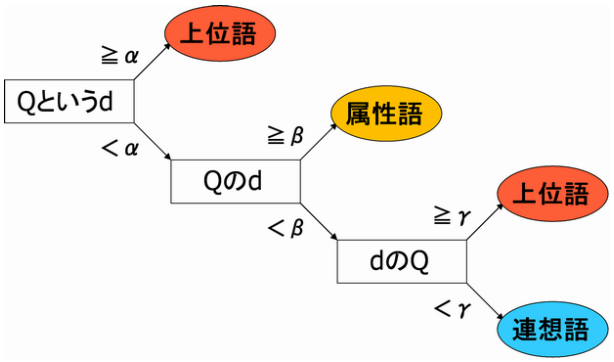

図 1: 現在の分類アルゴリズム

図 1 のように「クエリ＋統語パターン＋デスクリプタ」を検索表現として WWW 検索を行い，検出件数が閾値を越えるか否かによって「上位語」「属性語」「連想語」のいずれかに分類する．

桝井ら [2] は Murasaki のデスクリプタ分類性能の評価を行い，分類成功率 0.63 であったと報告している．この結果は一定の有効性を示すものであるが，現状はデスクリプタとして不適当な語を処理せずに分類したり，分類誤りが起こる場合がある．これではデスクリプタによるクエリの理解に支障をきたしてしまう．クエリ「Hotmail」を例に挙げると，デスクリプタ「フリーメール」は「上位語」,「フリーメールアドレス」は「属性語」に分類されるべきである．しかし，現状では「フリーメール」や「フリーアドレス」が「上位語」,「フリーメールアドレス」が「属性語」,「freemail」が「連想語」に分類されている．このように同義の語がそれぞれ別カテゴリに誤って分類される事象がある．

3　予備調査

本章では，Murasaki の分類精度向上のため，隅田らの手法を Murasaki に適用することについて議論する．隅田らは，Wikipedia の記事から定義文，カテゴリタグ，階層構造を利用して，大量の上位下位関係にある語の組を抽出する手法を提案している．

隅田らの手法を Murasaki のデスクリプタ分類機能に適用させるためには，抽出される上位下位データが Muraski 対象語[2]を保持している必要がある．そこで，隅田らの手法により抽出した上位下位データの，Murasaki 対象語に対する整合性を調査した．

以下，調査環境と調査結果について述べる．まず，調査環境について説明する．

上位下位データは，Wikipedia の定義文を用いる手法と，カテゴリタグを用いる手法の 2 通りを利用して抽出した．抽出のために利用した Wikipedia ダンプデータは 2011 年 10 月 23 日のものを用いた．

カテゴリタグを用いる手法により 1,661,555 組，定義文を用いる手法により 371,758 組の上位下位関係が得られ，合計で 2,003,724 組（重複を除く）の上位下位関係が得られた．

上位下位データに含まれる語の数は，カテゴリタグを用いる手法では 634,853 語，定義文を用いる手法では 396,163 語得られ，合計で 687,957 語（重複を除く）の語が得られた．

Murasaki 対象語として，Murasaki に実際に入力されたクエリ 228 語を用いた．また，クエリ 228 語中，上位下位データにエントリのあったクエリを対象に，デスクリプタの調査も行った．

デスクリプタは，Murasaki がシステム内部で付けたスコア[3]によるランキングの 1〜10 位まで，11〜20 位まで，21〜30 位までの 3 通り抽出して用いた．

整合性判断の指標として，上位下位データの語と Murasaki 対象語の一致率を用いた．一致率 S は以下の計算式により算出する．

$$S(\%) = \frac{C}{N} \times 100 \qquad (1)$$

N：対象語数

C：N の内，上位下位データと一致した数

一致しているか否かの判定基準として，Murasaki 対象語と上位下位データの語が完全一致する場合は「一致している」，そうでない場合は「一致していない」と判断した．

次に調査結果を報告する．

Murasaki 対象語にクエリを用いた整合性調査の結果では，カテゴリタグを利用する手法と定義文を利用する手法の，両手法を合わせた結果が 60.96%と最も高い一致率となった．今回の結果に加え，Wikipedia の階層構造を利用する手法を用いることにより，更に高い一致率が得られると考えられる．

[2]Murasaki は一般名詞の他に，固有名詞，新語，専門用語などを対象としたデスクリプタ抽出が可能であり，Murasaki 対象語は Murasaki によってデスクリプタ抽出が可能な語全てを指す．

[3]Murasaki は WWW から抽出したデスクリプタに，*pf*icf* 値や相対頻度を用いてスコアリングを行っている．スコアはクエリに対してのデスクリプタの妥当性を表している．

デスクリプタを用いた整合性調査の結果では，各手法の 1-10 位での結果で {28.06%，31.47%，31.82%} と最も高い一致率が得られた．スコアの低いデスクリプタは，クエリとは関係性の低い語や形態素解析誤りなどのノイズが多く含まれているため，スコア順位を広く設定したタスクでは一致率が上がらなかったと考えられる．この結果から，分類を行う際，スコア順位の高いデスクリプタのみで分類を行うことにより，分類精度が向上できると考えられる．

クエリを用いた調査結果は，隅田らの手法を Murasaki に適用することの有効性を示しているが，デスクリプタを用いた調査結果では一致率が上がらなかった．原因の一つとして，Murasaki が対象とする語の階層と，上位下位データが対象とする語の階層に差が生じている可能性が挙げられる．例えば，Murasaki 対象語の「ナス」は上位語として「野菜」がエントリされているが，上位下位データには上位語として「ナス属」がエントリされている．この結果から，Murasaki に対して上位下位データはより粒度が細かいといえる．

4 上位下位階層の洗練

本章では，上位下位階層を洗練することによる，Murasaki 対象語と上位下位データの整合性向上について述べる．

以下，4.1 節で基本的な考え方について述べ，4.2 節で実際のモデルについて説明する．

4.1 基本的な考え方

前章での予備調査の結果，上位下位データは Murasaki が対象とする語の階層より粒度が細かい可能性が示唆された．

この階層の差を解消する方法として，Stoica ら [5, 6] が提案している *Castanet Algorithm* を用いることを考えた．*Castanet Algorithm* は，WordNet の上位下位階層の必要な部分を取り出すことによって，階層構造を洗練する手法である．

Castanet Algorithm の処理概要を以下に示す．

1. コアツリー構築
   (a) コアツリーへの追加候補語の曖昧性判定
   (b) 非曖昧語のパスを調べ，コアツリーへ結合
2. コアツリー圧縮
   (a) k 個未満の子を持つ親を除去
   (b) 親の名前に現れる名前を持つ子を除去

この手法を，Murasaki の対象階層より細かい粒度の上位下位データに用いることで，Murasaki 対象語との整合性の向上が期待できる．前章の「ナス」を例に挙げて説明する．上位下位データで「ナス」に関連するコアツリーの構築を行うと上位階層から { 野菜-ナス属-ナス } となる．ここでコアツリー圧縮ルールの (b) を適用することにより，階層から「ナス属」が除去されて圧縮ツリーの { 野菜-ナス } が生成される．この上位下位関係は Murasaki が対象とする階層と一致するため，上位下位データと Murasaki 対象語の整合性が増すといえる．

また，応用として構築された上位下位階層を Murasaki のクエリとデスクリプタに適用することによって，階層を利用したデスクリプタ同士の類似語判別や，他クエリとの関連性の明示化などが期待できる．

4.2 上位下位階層洗練モデル

まず，コアツリー構築について考える．*Castanet Algorithm* は WordNet 内の各 synset [4]に関連付けられたテキストが存在することを前提としている．コアツリーへの追加候補となる語は，そのテキストのトピックを最も良く反映する語を選択する必要があるため語の曖昧性の判別が必要になる．しかしながら，上位下位データは幅広いドメインを持つ Murasaki 対象語との関連を持たせるため，語の曖昧性を考慮する必要は無い．

そこで，コアツリー構築には全上位下位データを対象として行う．上位下位データは，上位語と下位語の組で構成されているため，図 2 のようなコアツリー構築手法が利用できる．図 2 は，任意の上位下位関係の「上位語（下位語）」と全上位下位関係の「下位語（上位語）」を比較し，一致したら任意の上位下位関係の「下位語（上位語）」を比較した上位下位関係に「最下位語（最上位語）」として結合して，新しい階層の構築を行う様子を表している．新しい階層が生成されたら，図 2 の「new hierarchy」の処理を行う（新階層に対しても比較・結合を行う）．図 2 の手法の適用後，*Castanet Algorithm* と同様に共有パスの結合を行い，コアツリーを構築する．

次に，コアツリーの圧縮について述べる．コアツリー圧縮の 2 つの圧縮ルールには，WordNet 特有の制約が無い．そのため，上位下位関係から構築したコアツリーにコアツリーの圧縮ルールをそのまま適用できる

[4]WordNet の原単位で，個々の概念への同義語の集合．

```
for 全上位下位階層Hiに対して do
    Hiから上位ノードUiを抽出
    Hiから下位ノードLiを抽出
    for 全上位下位階層Hjに対して do
        if Hjの上位ノードとLiが一致 do
            新階層 N=Ui+Hj の生成
            goto new hierarchy
        end
        if Hjの下位ノードとUiが一致 do
            新階層 N=Hj+Li の生成
            goto new hierarchy
        end
    end
end

new hierarchy:
for i={0,...,i 1}のHiに対して do
    Hiから上位ノードUiを抽出
    Hiから下位ノードLiを抽出
    if Nの上位ノードとLiが一致 do
        新階層 N=Ui+N の生成
        goto new hierarhcy
    end
    if Nの下位ノードとUiが一致 do
        新階層 N=N+Li の生成
        goto new hierarhcy
    end
end
```

図 2: 上位下位データにおけるコアツリー構築手順

と考えられる．しかしながら，*Castanet Algorithm* が対象とする WordNet は英語で記述されたものであるため，日本語を対象とした上位下位データで適切に働くとは限らない．また，曖昧性を考慮していないことも影響する可能性が考えられる．そのため，実際に適用を行い，最終的に構築されたツリーの評価を行う必要がある．

5 おわりに

今回，Wikipedia から抽出した上位下位データを Murasaki に適用するために行った事前調査と，その調査から判明した問題解決のために *Castanet Algorithm* を拡張することを提案した．

予備調査の結果は，Murasaki 対象語としてクエリを用いた結果の一致率は 60.96%，デスクリプタを用いた結果の一致率は 31.82%であった．クエリの一致率から，上位下位データを Murasaki に適用させるための一定の有効性を示したが，デスクリプタでの調査で十分な一致率を得ることは出来なかった．原因の 1 つとして，Murasaki と上位下位データが対象とする上位下位階層に差が生じている可能性を挙げた．

そこで，我々は Stoica らが提案する *Castanet Algorithm* を上位下位データに適用することを提案し，具体的な手法について述べた．

現在，実際に上位下位階層洗練モデルの構築を行っている．今後は，実際に構築したモデルを実際に運用して評価を行っていく．また，応用として階層を利用したデスクリプタ同士の類似語判別や，他クエリとの関連性の明示化なども行っていきたい．

謝辞

本研究は科研費（基盤研究 (C) 20500833）による助成を受けたものである．

参考文献

[1] 川村佳史，桝井文人，河合敦夫，井須尚紀: ”WWW からの descriptor 抽出システムの開発とその評価”，言語処理学会第 13 回年次大会発表論文集，pp.1133–1136(2007.3)

[2] 桝井文人，ジェプカ・ラファウ，木村泰知，福本淳一，荒木健治: ”WWW 活用による語の比喩的素描手法”, 知能と情報，Vol.22, No.6, pp.707-719(2010.12)

[3] 長谷川恭佑，桝井文人: ”比喩的素描機構のための視覚化インタフェースの設計”，情報処理北海道シンポジウム 2011，pp.193–194(2011.10.1)

[4] 隅田飛鳥，吉永直樹，鳥澤健太郎，萬成賢太郎: ”Wikipedia からの大規模な上位下位関係の獲得”，言語処理学会第 14 回年次大会発表論文集，pp.769-772(2008.3)

[5] Stoica, E. and Hearst, M. Nearly-Automated Metadata Hierarchy Creation, in the Companion Proceedings of HLT-NAACL'04, Boston, May 2004.

[6] Stoica, E., Hearst, M., and Richardson, M., Automating Creation of Hierarchical Faceted Metadata Structures, in the Proceedings of NAACL/HLT 2007, Rochester, NY, April 2007.