

評判情報可視化のための評価表現辞書の有効性評価

谷本 融紀[†] 太田 学[†]

[†] 岡山大学大学院自然科学研究科 〒700-8530 岡山市北区津島中 3-1-1

E-mail: †{tanimoto,ohta}@de.cs.okayama-u.ac.jp

あらまし ネット上の評判情報は商品購入やマーケティングに有用だが、情報が膨大であるため全容把握が困難である。そのために評判情報の可視化に関する研究が現在多く行われており、我々は評価表現辞書を用いた評判情報の可視化を提案している。本稿では、評判情報の可視化に不可欠な自動作成する評価表現辞書の有効性を実験により検証したので、その結果について報告する。

キーワード 評判情報検索, 評価表現, 評価表現辞書, 評価値, 評価属性, 評価副詞, 点数化

Evaluation of Evaluative Expression Dictionaries for Visualizing Reputation

Yuki TANIMOTO[†] and Manabu OHTA[†]

[†] Graduate School of Natural Science and Technology, Okayama University

3-1-1 Tsushima-naka, Kita-ku, Okayama-shi, Okayama 700-8530, Japan

E-mail: †{tanimoto,ohta}@de.cs.okayama-u.ac.jp

Abstract Online reputation is useful for choosing products and marketing research. However, it is not easy to grasp reputational contents at a glance because they usually involve a vast amount of textual information. Therefore, there has been much research on visualizing reputation. We have proposed a mode of visualizing reputation using evaluative expression dictionaries. In this paper, we report the results of experiments for evaluating the proposed dictionaries to visualize reputation.

Key words reputation search, evaluative expression, evaluative expression dictionary, evaluative value, evaluative attribute, evaluative adverb, scoring

1. はじめに

近年では、評判情報がブログやレビューなどのテキスト情報として投稿されており、多数の意見を Web で閲覧することが可能である。しかしこれらの情報は通常テキストであるため、大量の内容を短時間で把握するのは困難である。またいつも限られた内容のみを閲覧するようになれば、情報に偏りが生じる恐れがある。そこで我々は特定の商品などについて、評判情報のテキストに基づいて評価を評価属性ごとにスコアで示し、可視化する手法を提案している。また、評判情報のテキストに現れる評価表現のスコアを計算するための評価表現辞書を自動で作成している。評価表現は「良い」「悪い」などの評価値、「デザイン」「操作性」などの評価属性、「かなり」「とても」などの評価副詞の三つから構成されるとし、これらを組み合わせると評価表現辞書は構成される。また評価値の極性は評価属性に依存するので、評価値は評価属性とのペアで扱う。本稿ではこの評価表現辞書の有効性を検証し、課題について考察する。

2. 関連研究

評判の抽出に関して、あらかじめ人手で評価表現辞書を作成し、この辞書を用いて意見を抽出し提示する立石らの研究 [1] がある。評価表現辞書には「書籍」「コンピュータ」などのドメインごとに、評価値と「肯定」「否定」の情報を登録する。一方評判抽出では、はじめに Web から評判情報の書かれたテキストを取得する。テキストの文中で、ユーザの入力した商品名と、評価表現辞書に登録された評価表現が十分近い距離にあるか判定する。そして評価表現として適正かどうか判定するルールを用い、条件を満たした文を評価文とみなす。この評価文を肯定か否定に分類し、ユーザに提示する。立石らの研究では、辞書を人手で作成するので高品質な辞書の作成が可能であるが、ドメインごとに大規模な辞書を作成するのは容易でない。また提示する情報は、ユーザの登録した商品がレビューテキストにおいて高評価か低評価かの評価極性だけであり、評価属性ごとに評価するものではない。

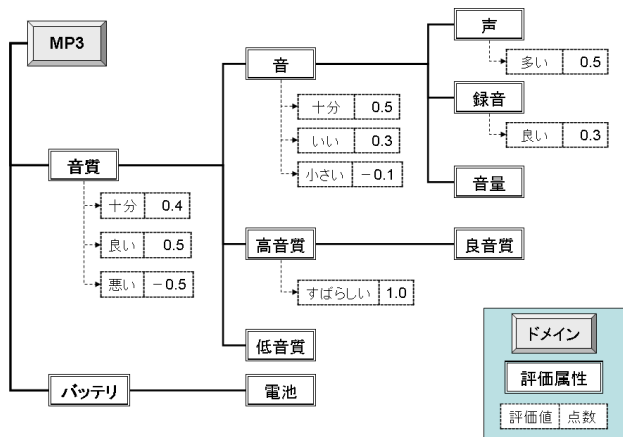


図 1 ディレクトリ構造をもつ評価属性-評価値辞書の一例

Web 上のレビューを肯定、否定に分類し評価表現辞書を自動で作成する研究 [2], [3] も行われている。藤村ら [3] が利用した価格.com [8] のレビューには商品ごとに「良」「悪」のようなタグをつけることができる。藤村らはこのタグを基に評価表現を肯定か否定かに分類し、出現頻度に基づくスコア付けを行っている。本研究でも価格.com を利用するが、肯定か否定かの極性ではなく、価格.com の評点から評価値の極性値を算出する点が異なる。また藤村らは商品の評価属性を考慮していない。

評価属性を抽出し、それをディレクトリ構造として上位属性、下位属性に分類する関口の手法がある [4]。これは商用検索エンジンに入力されたクエリの遷移を抽出し、絞込み検索に用いている語句を下位属性として登録する手法である。具体的には「ギャンブル」「地名」などクラスを用意し、「ギャンブル」に対しては「パチンコ」「宝くじ」などシードインスタンスを用意する。シードを用いて絞り込もうとしていた単語がそのクラスに対して偏りなく出現していれば、この単語をそのクラスの下位属性とする。

3. 評価表現辞書

3.1 評価文

本研究では、以下の要素により構成されるものを評価文と定義する。

- 評価主体 … 評価を行った主体を指す表現
- 評価対象 … 製品など特定のクラスの実体を指す表現
- 評価表現 … 評価対象に対する評価を表す表現
 - 評価値 … 評価対象や評価属性に対する評価を示す表現
 - 評価属性 … 評価対象の属性（側面）を示す表現
 - 評価副詞 … 評価値を強めたり弱めたりする表現

具体的にいえば、評価主体はレビューサイトにおけるレビューを指し、評価対象はドメインが「mp3」であれば「ipod touch」などがそれにあたる。またドメイン自体も評価対象になりえる。評価表現は評価値、評価属性、評価副詞からなり、評価値は「高い」「低い」など、評価属性は「デザイン」「音質」など、評価副詞は「あまり」「かなり」などがある。

3.2 評価表現辞書の構成

評価値の一部は評価属性に依存して極性が変わる。例えば、評価値「高い」は、評価属性が「音質」の場合は一般的に良い評価となり、評価属性が「値段」の場合は悪い評価となる。よって評価属性と評価値を組み合わせる必要がある。また、評価属性の中には関連性が高いものが存在する。例えば評価属性「音質」や「音」、「声」などは関連があると考える。よって本研究では評価属性の関連も考慮する。

そこで、本研究では図 1 に示すようなディレクトリ構造をもつ評価属性-評価値辞書を提案する [6]。

図 1 において、ドメインは「mp3」であり、その直下の評価属性として「音質」と「バッテリー」が登録されている。本稿ではこれら評価対象直下の属性を主属性と呼ぶ。さらに「音質」の評価値には「十分」「良い」などがある。また「音質」の関連属性として、「音」「高音質」「低音質」がある。さらに「音」の評価値には「十分」などがあり、関連属性には「声」などがある。なお評価値に付されている数値は $[-1, 1]$ の極性値であり -1 に近いほど不評、 1 に近いほど好評を表す。

評価副詞はこの評価値の極性値を増減する表現とみなすので、評価副詞辞書には副詞とその重みを登録し、重みの範囲は $[0, 2]$ とする。評価副詞は評価属性、評価値に比べ数は少なく、両者との依存関係も認められないため、評価副詞辞書は評価属性-評価値辞書とは別に手作業で作成する。

3.3 評価属性-評価値辞書の作成

評価属性-評価値辞書の作成には価格.com のレビューテキストと、それに付与されている評点を利用する。作成の流れを図 2 に示す。

図 2 の (1) の二つ組の生成手順は 3.3.1 項で (2) のディレクトリ構造を持つ辞書の構築方法については 3.3.2 項で (3) の辞書三つ組の生成方法は 3.3.3 項で説明する。

3.3.1 評価属性、評価値の二つ組の生成

図 3 は価格.com にあるレビューの例で、本研究で利用する「項目」と「評点」、さらに「評価属性候補」と「評価値候補」を示している。このようなレビューを利用して評価属性と評価値の二つ組、およびそれに極性値を加えた三つ組である辞書三つ組を作成する。以下二つ組生成までの手順について説明する。

- (1) レビューテキストを価格.com より取得
- (2) レビューテキストから評価値を取得
- (3) 評価値にかかる評価属性を取得し二つ組を生成

本研究では価格.com のレビューテキストの取得には商品検索 API [8] を利用する。評価値には CaboCha [9] で解析した形容詞と形容動詞を、評価属性には名詞を抽出し、評価値にかかる評価属性を取得することで「評価属性、評価値」の二つ組を生成する。抽出作業の際、CaboCha の情報をもとに記号、名詞および形容詞の連結処理を行う [6]。連結処理は 2 語以上の形態素を一つに繋げる処理で、この連結処理により、例えば「扱う」「やすい」を「扱いやすい」にしたり、「難しい」「ない」を「難しくない」とすることができる。

3.3.2 ディレクトリ構築

二つ組の評価属性と図 3 に示した価格.com のレビューの「項

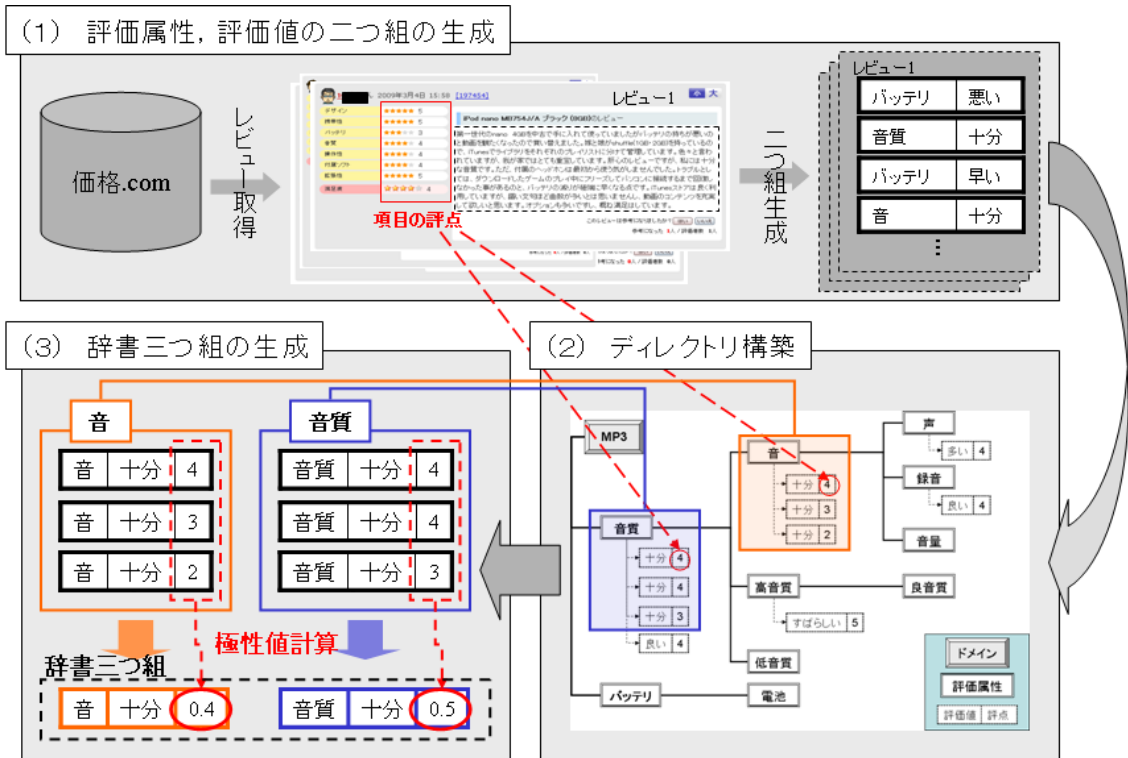


図 2 評価属性-評価値辞書の作成

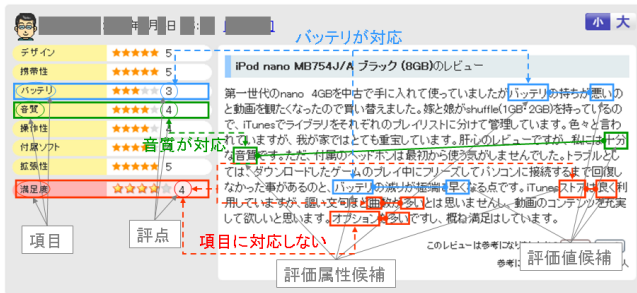


図 3 価格.com のレビュー

を生成したあと、評価属性の先祖の主属性ノードに対応する図 3 の「項目」の「評点」を各二つ組に付加し、仮の三つ組「評価属性、評価値、評点」を生成する。また主属性に対応付けられなかった場合は、図 3 の「項目」の満足度の「評点」を付加し、このような評価属性は独自属性と呼ぶ。この独自属性は主属性やその子孫属性ではないが、出現頻度が高い評価属性が選択されており、ドメインを特徴付けるものである。独自属性についても主属性と同様にしてその子孫属性からなるディレクトリを生成する。

3.3.3 辞書三つ組の生成

評価値の極性値は、各属性のノードに割り当てられた「評価属性、評価値、評点」の仮の三つ組を用いて評価属性ごとに計算する。

式 (1) で評価値 i の平均評点 $average_i$ を計算する。ここで $point_{i,k}$ はその評価属性において評価値 i を持つ k 番目の仮の三つ組の評点であり、 $point_{i,k} \in \{1, 2, 3, 4, 5\}$ である。また N_i はそのような仮の三つ組の総数である。

$$average_i = \frac{1}{N_i} \sum_{k=1}^{N_i} point_{i,k} \quad (1)$$

この $average_i$ を式 (2) を用いて $[0, 1]$ に正規化する。なお式 (3) (4) で n はその評価属性のもつ評価値の異なり数である。

$$normal_i = \begin{cases} \frac{average_i - m}{M - m} & (if M > m) \\ \frac{average_i - 1}{4} & (if M = m) \end{cases} \quad (2)$$

目」を対応付け、図 1 のようなディレクトリ構造を持つ評価表現辞書を生成する。図 4 はそのディレクトリ生成の擬似コードである。ここで 2 行目の $depth$ はディレクトリの深さを表し、5 行目の $Main_Attributes$ は図 3 の満足度を除く「項目」の集合である。append_child_node メソッドは自ノード直下に新たに子ノードを一つ生成して二つ組を配置し、append_child_nodes は子ノードを複数生成する、get_child_nodes は自ノードの子ノードを全て取得するメソッド、assign は自ノードに二つ組を配置するメソッドである。“ n . 評価属性”や“ p . 評価属性”はそれぞれの評価属性を取得することを意味する。equals メソッドは同一文字列の場合に true を返し、contains メソッドは文字列を包含している場合に true を返す。synonym メソッドは類語の場合に true を返すメソッドである。この類語は類語辞典 weblio [10] により取得した表現である。また remove は指定された要素を削除するメソッドである。

図 4 のアルゴリズムで、評価属性に対するディレクトリ構造

```

1 void gen_mainAttribute_directory(){
2     depth = 2;
3     二つ組集合 P;
4     ルートノード R;
5     主属性ノード集合 Main_Attributes;
6     R.append_child_nodes(Main_Attributes);
7     gen_directory(R.get_child_nodes(), P, depth);
8 }
9 void gen_directory(N, P, depth){
10    if(depth > 0){
11        for(n : N)
12            for(p : P)
13                if((n. 評価属性).equals(p. 評価属性)){
14                    n.assign(p);
15                    P.remove(p);
16                }
17        for(n : N)
18            for(p : P)
19                if((n. 評価属性).contains(p. 評価属性) ||
20                    (p. 評価属性).contains(n. 評価属性) ||
21                    (n. 評価属性).synonym(p. 評価属性)){
22                    n.append_child_node(p);
23                    P.remove(p);
24                }
25        for(n : N)
26            gen_directory
27                (n.get_child_nodes(), P, depth-1);
28    }
29 }

```

図 4 主属性とその子孫属性のディレクトリ生成の擬似コード

$$M = \max_{i=1, \dots, n} average_i \quad (3)$$

$$m = \min_{i=1, \dots, n} average_i \quad (4)$$

そして式 (5) により評価値 i の極性値 $value_i$ を算出し、その評価属性に対応する極性値として割り当て、評価属性-評価値辞書に格納する辞書三つ組「評価属性、評価値、極性値」とする。

$$value_i = \begin{cases} shift_i & (if\ shift_i > 0) \\ shift_i / 5 & (otherwise) \end{cases} \quad (5)$$

ただし式 (5) の $shift_i$ は以下の式 (6) で計算する。

$$shift_i = 6 \cdot normal_i - 5 \quad (6)$$

レビューテキストでは「悪い」などと表現されていても、評点の 5 段階評価で「3」など悪くない評点を与えられている場合がしばしばある。このような評価値に対するバイアスを補正するために式 (6) を用いて $shift_i$ を求める。 $shift_i$ のままでは $[-5, 1]$ となるため、式 (5) によって $value_i$ を $[-1, 1]$ にする。

4. 提案システム

本章では評価表現辞書を用いた評価表現のスコア計算と提案システムについて述べる。

4.1 評価表現辞書を用いた評価表現のスコア計算

レビューテキストの可視化の概略を図 5 に示す。可視化するレビューの取得には価格.com 商品検索 API や Yahoo!JAPAN のウェブ検索 WebAPI [11] を用いる。取得するレビューの件数はユーザが指定できる。そして各レビューテキストについて属性ごとのスコアを計算し、このスコアを用いてランキングする。評判情報可視化の流れを図 5 を使って説明する。

- (1) テキストを CaboCha を用いて係り受け解析する
- (2) 評価表現三つ組を生成する
- (3) 評価表現三つ組のスコアを計算する
- (4) 各評価属性に対するスコアを計算する
- (5) 各主属性のスコアを計算する
- (6) 総合評価のスコアを各主属性のスコアの平均とする

評価値、評価属性の取得は 3.3.1 項と同様に行い、連結処理も同時に行う。評価副詞は評価値に係る副詞を探索することで取得し、「評価属性、評価値、評価副詞」の評価表現三つ組を生成する。そして評価属性-評価値辞書と評価副詞辞書の両方を用いて評価表現三つ組のスコアを計算する。このスコアの値は式 (7) により計算する。

$$score = value \cdot adverb \cdot attribute \quad (7)$$

ここで $value$ は評価表現三つ組の「評価属性、評価値」に対応する評価属性-評価値辞書の極性値、 $adverb$ は評価副詞辞書にある評価副詞の重みで、評価副詞がなければ 1 とする。 $attribute$ は特定の属性をどの程度重視するかに応じて与える値が変化し、指定がない場合は 1 とする。そして取得した全ての評価表現三つ組のスコアを式 (7) を用いて計算し、同じ評価属性をもつ評価表現三つ組ごとにその平均を求め、各評価属性のスコアとする。

次に主属性のスコアを計算する。そのために主属性と評価属性との対応付けを行う。評価属性が主属性でない場合、評価属性-評価値辞書においてその先祖属性に主属性がないか探索し、あればその主属性に対応付ける。主属性と、それに対応付けられた評価属性のスコアの平均を、各主属性のスコアとする。さらに全主属性のスコアの平均を総合評価のスコアとする。そして評価属性と主属性、および総合評価のスコアをランキングに用いる。

評価属性が独自属性である場合も主属性の場合と同様に「評価値、評価属性、評価副詞」の評価表現三つ組を生成したのち、式 (7) などを用いて各独自属性のスコアを計算する。

4.2 評判情報の可視化

4.1 節で求めた評価属性のスコア、主属性のスコア、および総合評価のスコアとレビューテキストを可視化に利用する。図 6 はレビューテキストの可視化の例である。この提案システムでは、ドメインとレビュー対象の商品名、検索件数、サイトの

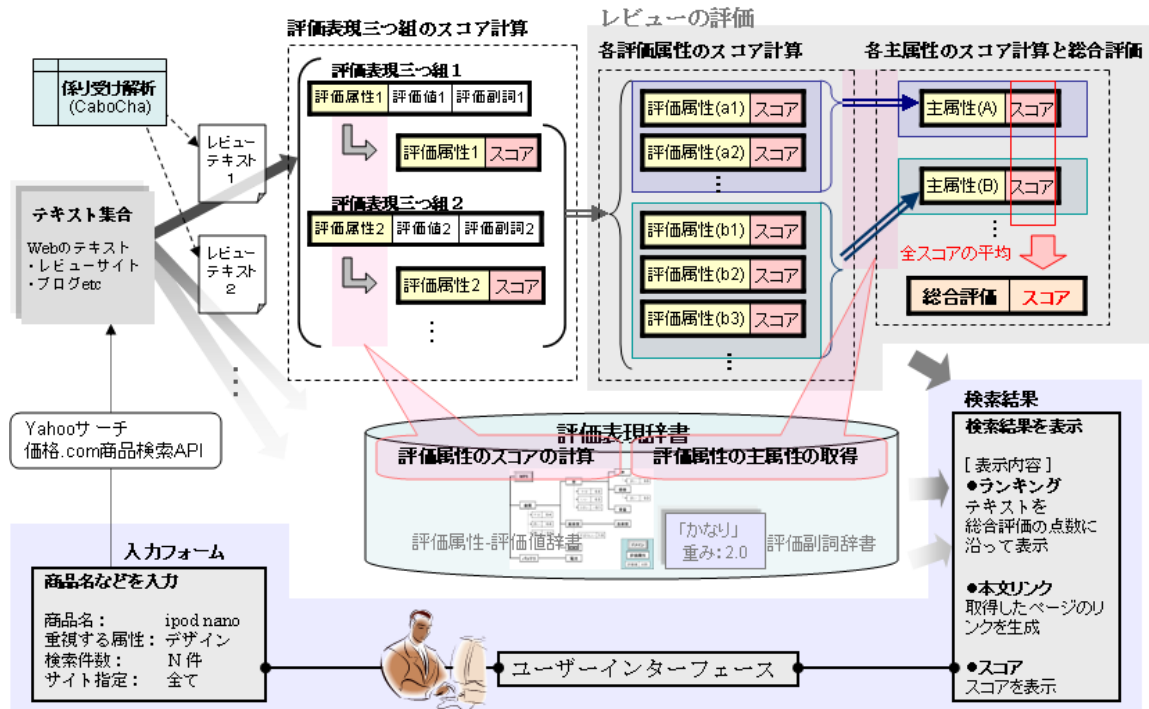


図 5 レビューテキストの可視化

表 1 実験で使用したレビューテキスト

日付	キーワード	カテゴリ	テキスト数
2010 月 10 月	mp3	Pc	1839
2010 月 12 月	ノートパソコン, MacBook	Pc	1856
2010 月 12 月	ゲーム	Game	2614

表 2 各ドメインの主属性

ドメイン	mp3	ノートパソコン	ゲーム
主属性	デザイン 携帯性 バッテリー 音質 操作性 付属ソフト 拡張性	デザイン 処理速度 グラフィック性能 拡張性 使いやすさ 携帯性 バッテリー 液晶	操作性 グラフィック サウンド 熱中度 継続性 ゲームバランス

表 3 各主属性の子孫属性の正解率

カテゴリ	mp3	ノートパソコン	ゲーム
子孫属性の数	158	145	92
正解率 (rigid)	0.513	0.566	0.651
正解率 (relaxed)	0.867	0.951	0.965

指定を行い、レビューを検索する。ここで重視する属性にチェックを入れると、チェックを入れた主属性とその子孫属性の重みを大きくことが出来る。そしてテキストを解析し、その結果を画面右側に出力する。図 6 の出力結果の最上部に商品の評価を出力する。その下は解析したレビューテキストを任意の主属性のスコアの昇順、または降順にソートして表示するボタンである。さらにその下の左側には、各レビューテキストに対する評価属性ごとのスコアと総合評価のスコア、右側にはレビューテキストを表示している。左の評価属性は上から順に主属性、独自属性と並んでいる。

5. 評価表現辞書の評価

実験では作成した評価属性-評価値辞書と、可視化したレビューテキストのスコアを評価する。辞書作成のためのコーパスとして用いたレビューテキストは、価格.com の商品検索 API を利用して取得した。取得した時期と、この API に入力したキーワード、カテゴリ、また取得したレビューテキストの数を表 1 に載せる。またここで用いたキーワードをドメインとして、ドメインごとの主属性を表 2 に示す。ただし「ノートパソコン, MacBook」のドメインは「ノートパソコン」とした。そして評価属性-評価値辞書をドメインごとに作成した。

5.1 主属性の子孫属性の適切性の評価

まず各ドメインの主属性の子孫属性について適切性を評価す

る。表 3 は各ドメインの主属性に対する子孫属性の正解率である。「子孫属性の数」は表 1 のレビューテキストを解析することで取得した各主属性の子孫属性の総数である。「正解率 (rigid)」は、主属性との関連が明らかな子孫属性だけを正解とした正解率である。「正解率 (relaxed)」は正解判定を緩め、文脈によっては主属性との関連が認められる子孫属性も正解とした正解率である。また表 4 と表 5、表 6 にそれぞれ「mp3」「ノートパソコン」「ゲーム」に対する評価属性の正解分類例を示す。例えば表 4 の主属性「デザイン」に対する rigid 判定における正解は「形」「外観」など、relaxed 判定における正解は「変形」「姿勢」などであり、「言い方」は子孫属性ではないことを示す。表 5 と表 6 についても同様である。またドメインの特徴を表す独自属性の例を表 7 に示す。なお独自属性の総数は「mp3」で

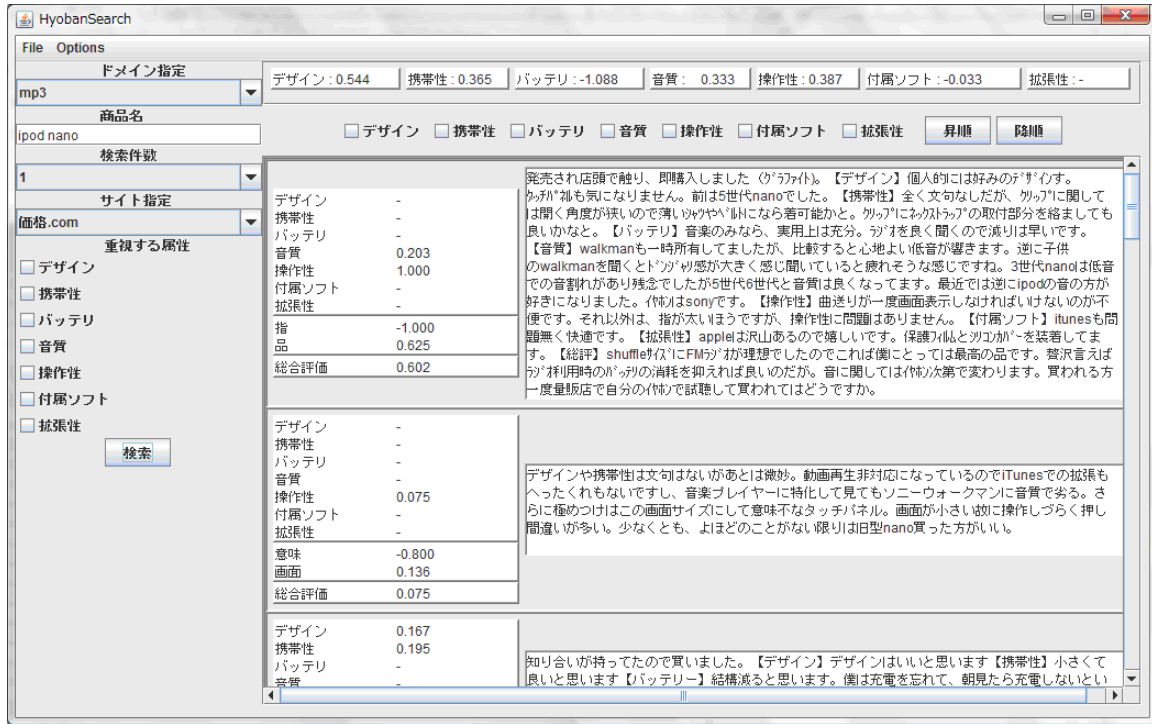


図 6 提案システムの実行画面

表 4 「mp3」の評価属性の分類例

主属性	デザイン	携帯性	バッテリー	音質	操作性
子孫属性 (rigid)	形 外観	携帯 持ち運び	バッテリー 電池	声 低音	操作 扱い
子孫属性 (relaxed)	変形 姿勢	-	寿命 容量	程度 基本音質	運用 工夫
子孫属性ではない	言い方 -	-	ノイズ ホワイト	思い ただ	-

表 5 「ノートパソコン」の評価属性の分類例

主属性	デザイン	処理速度	使いやすさ	液晶
子孫属性 (rigid)	格好 模様	高速 スピード	使い道 使い勝手	画面 スクリーン
子孫属性 (relaxed)	やり方 -	対応 解決	使い 使いで	光沢 ワイド
子孫属性ではない	地形 -	-	-	あと 自体

は 1560、「ノートパソコン」は 986、「ゲーム」は 2085 である。

「ノートパソコン」や「ゲーム」のドメインには表 2 に示すように、主属性として「グラフィック性能」や「サウンド」などが含まれている。この場合、評価属性「画質」などは「グラフィック性能」に対する子孫属性に、評価属性「音」「音質」などは「サウンド」に対する子孫属性になることが期待されるが、類語や類似文字列として取得できなかったために独自属性に分類されていた。これは主属性に対する子孫属性の網羅性の観点からは問題であり、検討の余地がある。

また主属性の「グラフィック性能」の文字列の一部である「性能」により、評価属性「性能」が「グラフィック性能」の子属性となっていた。この「性能」は「グラフィック」に限定されない評価属性であるので、別の属性の子孫にも含まれることが望ましい。

5.2 辞書三つ組「評価属性、評価値、極性値」の評価

ここでは評価属性-評価値辞書辞書の評価属性に結び付けられた評価値の極性値が妥当かどうかという観点から、「評価属性、評価値、極性値」の辞書三つ組を評価する。評価対象は主属性と 5.1 節において、rigid 判定で正解としたその子孫属性のみとする。

表 6 「ゲーム」の評価属性の分類例

主属性	操作性	グラフィック	サウンド	熱中度
子孫属性 (rigid)	操作 制御	瀕死グラフィック グラフィックレベル	サウンドカード エレキサウンド	熱中 -
子孫属性 (relaxed)	仕事 工夫	性能 レベル	サウンド同上 -	- -
子孫属性ではない	経営 -	-	画質サウンド 画質	虫 -

表 7 独自属性の例

ドメイン	mp3	ノートパソコン	ゲーム
独自属性	イヤホン メモリアイブ フラッシュメモリ スピーカー性能	省スペース スタンバイ マウスパッド リカバリ	動き バトル オンライン コントローラー

表 8 は生成した辞書三つ組の総数と正解率である。辞書三つ組の総数は取得した辞書三つ組の評価属性が主属性、または明らかに主属性の子孫属性であるものの数といえる。「正解率 (rigid)」はその評価属性と組み合わせたととき、一般的に良いと考えられる評価値の極性値が正、悪いと考えられる評価値の極性値が負であれば正解とした場合の正解率である。「正解率

表 8 「評価属性，評価値，極性値」の辞書三つ組の正解率

ドメイン	mp3	ノートパソコン	ゲーム
辞書三つ組の総数	1248	645	556
正解率 (rigid)	0.602	0.500	0.507
正解率 (relaxed)	0.795	0.730	0.757

(relaxed) は評価者によっては正解となるものも含めた正解率である．例えば「音質，いい，0.49」という辞書三つ組を考える．ここで評価属性「音質」と評価値「いい」を組み合わせると，明らかにこれらは良い評価を表している．そして極性値は正の値である．よってこの辞書三つ組は rigid 判定で正解となる．しかし「デザイン，ダサイ，1.0」の場合「デザイン」が「ダサイ」のは明らかに悪い評価であるのに極性値が正の値であるのでこれは誤りである．一方で「音，近い，-0.4」などこれらだけでは妥当かどうか判断できないものは relaxed 判定で正解としている．

実験で relaxed 判定で正解とした辞書三つ組の中には，評価者によっては rigid 判定でも正解となるものが存在した．例えば「評価属性，評価値」が「高音，強い」のとき，極性値が正のときは高音が強いのを好む人にとっては rigid 判定で正解となるが，弱い方を好む人にとっては誤りであると考えられる．relaxed 判定の正解にはこのような例も含まれる．

また実験で用いた辞書三つ組の評価属性だけでは評価値の極性値の適切性を判断できないものもあった．例えば「バッテリー，長い」という評価属性・評価値の組では「バッテリー」の「持ち」が「長い」のか「バッテリー」の「充電時間」が「長い」のかで極性が異なる．また「バッテリー，早い」なども同様である．よって辞書の構築において，二次属性を考慮するなどの工夫が必要である．

5.3 可視化したレビューテキストのスコア

提案した評価表現辞書と，それを利用したレビューテキストの可視化システムによる可視化の適切性を，三つのドメイン「mp3」「ノートパソコン」「ゲーム」の辞書を用いて確認する．4.1 節の評価表現三つ組のスコアと，レビューテキストに付随している項目の評点の比較を行った．これらの相関を「mp3」，「ノートパソコン」，「ゲーム」についてそれぞれ図 7，図 8，図 9 に示す．これらの図中の一つの点が，評価表現三つ組のスコアとレビューテキストの評点に対応している．すなわち横軸は図 3 の価格.com におけるレビューの評点，縦軸は提案するシステムにより計算した評価表現三つ組のスコアであり，直線は回帰直線である．これらの図において点の回帰直線からのはずれが小さければ提案手法が妥当であると考えられる．さらに評価対象となった主属性ごとの評価表現三つ組の総数と相関係数を「mp3」，「ノートパソコン」，「ゲーム」についてそれぞれ表 9 と表 10，表 11 にまとめる．これらの相関係数が高いほど，本稿の提案は妥当であると考えられる．

表 9 より「mp3」は評価属性により相関に大きく差があるのが分かる。「バッテリー」や「操作性」，「付属ソフト」や「拡張性」は相関係数が 0.3 以上であり，他の評価属性より比較的相関が大きいといえる．また「デザイン」「音質」にもそれらに

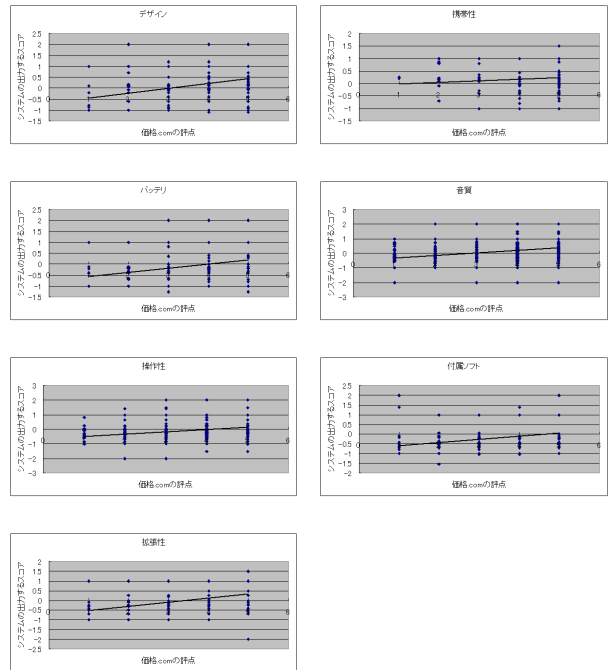


図 7 「mp3」の価格.com の評点と評価表現三つ組のスコアの相関

つづくの正の相関があった．一方で「携帯性」については相関があまりないという結果となっている．この理由については，係り受け解析の失敗や，不適切な連結処理がその一因であると考えている．

一方ドメイン「ノートパソコン」と「ゲーム」については表 10，表 11 より，全体的に正の相関が低い．これは主属性の子孫属性が十分取得できないために評価対象の数が不足していること，また誤った子孫属性が存在しているために不適切な評価対象を用いて相関係数を計算していること，そして評価表現辞書による評価表現三つ組のスコアの計算精度に原因があると考えられる．よって主属性に対する適切な子孫属性を多く取得し，それらに対する評価値の極性値を計算することが必要である．

6. まとめ

本稿では評判情報可視化のための評価表現辞書の作成法とその評価について述べた．評価表現辞書の作成では，価格.com のレビューテキストを CaboCha で解析し「評価値，評価属性」のペアを取得した．これを用いて評価属性に対するディレクトリ構造を生成し，価格.com に付与された評点を利用することで「評価値，評価属性」のペアに対する極性値を計算して「評価値，評価属性，極性値」の辞書三つ組を生成した．そして実験ではドメイン「mp3」，「ノートパソコン」，「ゲーム」に対して作成した評価表現辞書について，主属性に対する子孫属性の適切性と辞書三つ組の妥当性を評価した．また各ドメインについて，価格.com のレビューの項目ごとの評点と提案システムで計算した評価表現三つ組のスコアの相関図を各主属性ごとに示し，相関係数を計算した．その結果，一部の主属性とその子孫属性については相関係数が 0.3 以上の正の相関が認められた．一方で無相関に近い主属性とその子孫属性もあった．よって今

表 9 「mp3」の主属性の相関係数

主属性	デザイン	携帯性	バッテリー	音質	操作性	付属ソフト	拡張性
評価表現三つ組の総数	1106	581	295	2810	1052	281	188
相関係数	0.264	0.102	0.329	0.284	0.305	0.319	0.436

表 10 「ノートパソコン」の主属性の相関係数

主属性	デザイン	処理速度	グラフィック性能	拡張性	使いやすさ	携帯性	バッテリー	液晶
評価表現三つ組の総数	1264	1266	1263	494	138	959	423	1708
相関係数	0.230	0.207	0.181	0.197	0.248	0.241	0.272	0.167

表 11 「ゲーム」の主属性の相関係数

主属性	操作性	サウンド	グラフィック	熱中度	継続性	ゲームバランス
評価表現三つ組の総数	611	252	899	131	99	1020
相関係数	0.321	0.261	0.211	0.301	0.285	0.250

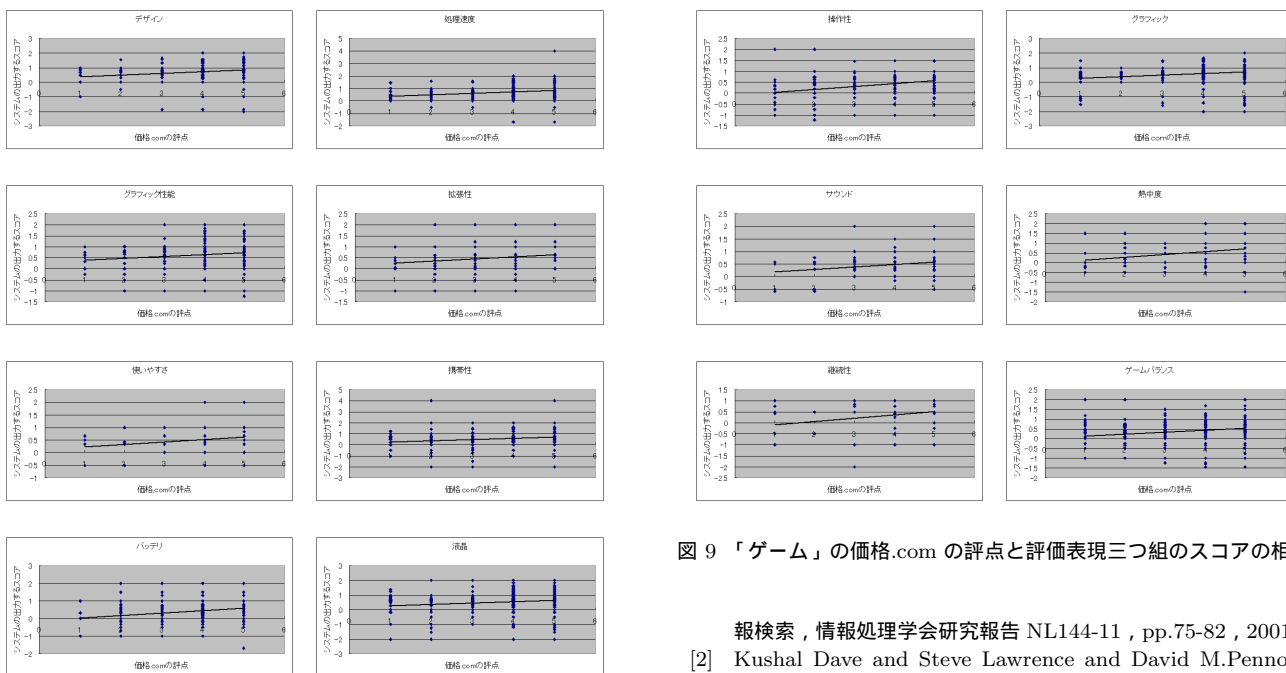


図 8 「ノートパソコン」の価格.com の評点と評価表現三つ組のスコアの相関

後辞書の構築方法やスコア計算の方法についてさらに検討する必要がある。

主属性に対する子孫属性の適切性についての評価実験では適合率のみを計算したが、主属性に対する子孫属性の網羅性の観点から、取得した全評価属性の分類の妥当性も評価したい。また価格.com 以外のレビューサイトの可視化や可視化の処理コスト、および独自属性の適切性などについての評価も今後行う予定である。

文 献

[1] 立石 健二, 石黒 義英, 福島 俊一: インターネットからの評判情

図 9 「ゲーム」の価格.com の評点と評価表現三つ組のスコアの相関

報検索, 情報処理学会研究報告 NL144-11, pp.75-82, 2001.

[2] Kushal Dave and Steve Lawrence and David M.Pennock: Mining the Peanut Gallery: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, International World Wide Web Conference(WWW2003), pp.519-528, 2003.

[3] 藤村 滋, 豊田 正史, 喜連川 優: 電子掲示板からの評価表現および評判情報の抽出, JSAI2004 3F1-03, 2004.

[4] 関口 裕一郎, 田中 智博, 内山 匡, 藤村 滋, 望月 崇由, 鈴木 智也: 検索クエリログのセッション情報を利用した属性語句抽出, DEIM2010 A2-3, 2010.

[5] Amazon
<http://www.amazon.co.jp/>

[6] 谷本 融紀, 太田 学: 評価属性を考慮した評判情報の可視化, 情報処理学会研究報告, Vol.2010-DBS-151, No.14, pp.1-8, 2010.

[7] NLPProcessor Text Analysis Toolkit, 2000,
<http://www.infogistics.com/textanalysis.html>

[8] 価格.com,
<http://kakaku.com/>

[9] 日本語係り受け解析器 CaboCha,
<http://chasen.org/~taku/software/cabocha/>

[10] 類語辞典 Weblio,
<http://thesaurus.weblio.jp/>

[11] Yahoo!JAPAN のウェブ検索 WebAPI,
<http://developer.yahoo.co.jp/webapi/search/websearch/v1/websearch.html>