

特定評価属性の関連属性自動抽出による評価表現辞書の生成

谷本 融紀^{1,a)} 太田 学^{1,b)}

概要: 我々は商品の評判情報を特定の評価属性毎に数値化し可視化する評判情報検索システムを提案している。提案システムでは評価表現辞書を用いるが、この辞書はレビューテキストを利用して自動生成する。このとき特定の評価属性毎に数値化するために、それらに関連する評価属性である関連属性を抽出し収集する必要がある。本研究ではレビューテキストを適当な粒度に分割し、各評価属性の共起頻度を用いてこの関連属性を自動抽出した。また実験により、このようにして取得した関連属性の適切性と、生成した評価表現辞書を用いて計算した評判情報の評価極性値を評価した。

Construction of an Evaluative Expression Dictionary by Extracting Attributes Related to Specific Evaluative Attributes

YUKI TANIMOTO^{1,a)} MANABU OHTA^{1,b)}

1. まえがき

Web上の評判情報から必要な情報のみを抽出し、分かりやすく提示する試みが現在多く行われている。そこでは評判情報を肯定極性、否定極性の二値に分類する研究が多いが、単純に分類できない場合が存在する。例えば評価値「良い」と「すごく良い」では、肯定の度合いが異なり、「普通」はどちらの極性にも属さない。また、評価値は対象の「デザイン」などの評価属性について述べられている場合が多い。

そこで本研究では、レビューテキストを用いて、評価属性ごとに分類した辞書極性値付きの評価表現辞書を自動生成する。我々はこの評価表現辞書を用いて、評判情報を評価極性値に変換し、評価属性ごとに可視化することを目指している。本稿では適当な粒度に分割したレビューテキストを用い、共起頻度に基づいて評価属性の階層構造を生成する。そしてこの階層構造を評価表現辞書に反映する。

本論文の構成を以下に示す。2節で関連評価属性の収集と評判情報検索の関連研究について説明し、3節で本研究

における評判情報の定義を述べる。4節では評価属性の関連付けと評価表現辞書について、5節で評価表現辞書を利用した評価極性値の計算方法について説明する。そして6節で評価実験について述べ、7節でまとめる。

2. 関連研究

2.1 評価属性の関連付け

西田ら [1] は、「教えて!goo」^{*1}のQ&A文書を用いてドメインと、そのドメインに関連する名詞である主題タグとキーワードを用いて3階層のタギングを行うシステムを提案している。主題タグはキーワードより、ドメインは主題タグより意味的抽象度が高い。ドメインはQ&Aコミュニティのカテゴリ名を用い、ドメインとそのタイトル中に出現する名詞との共起確率をFisher's Exact Test[2]を用いて検定し、有意な場合にその名詞を「主題タグ」とした。またTFIDF法に基づいてキーワードを選出した。これは特定のドメインの下位概念を構成する点で本稿での提案と類似する。しかし、主題タグとキーワードの名詞同士の関連付けを行っていない点は異なる。

関口ら [3] は検索クエリの遷移を利用することでドメイン同士を関連付けている。あらかじめ「地名」や「駅」など抽象度の高い名詞集合であるクラスを用意し、それらに

¹ 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University

a) tanimoto@de.cs.okayama-u.ac.jp

b) ohta@de.cs.okayama-u.ac.jp

^{*1} <http://oshiete.goo.ne.jp/>

関連する名詞であるシードインスタンスを数個用意する。例えば「駅」に対する「東京駅」「新宿駅」などがシードインスタンスである。そしてシードインスタンスによる検索結果を、ユーザがさらに絞り込むのに用いた語句を下位ドメイン候補とする。すなわち「東京駅」→「東京駅 構内図」と検索を変えた場合、「構内図」が「東京駅」に対する下位ドメイン候補となる。そして、全てのシードインスタンスにおいて偏りなく用いられているドメイン候補をクラスの下位語句と決定する。一方本研究は、ドメインの結び付けではなく、評価属性の関連付けが目的であるため異なる。また抽出対象がクエリ遷移である点も本研究と異なる。また山田ら [4] は決定木学習アルゴリズムの一つである C4.5[5] を利用して評価属性を分類している。

2.2 評価表現辞書

Kamps ら [6] は語彙ネットワークを用いて評価表現辞書を半自動で生成した。この方法は既存のシソーラスなどに依存するため、最新の評価表現や未知の評価表現に対応ができない点が問題である。

また「肯定極性をもつ表現の周辺文脈には肯定極性が現れやすい」という仮定に基づき、肯定極性と否定極性それぞれの種表現を用意し、それらと共に起る表現の評価極性を判定する Turney[7] の研究がある。さらに接続詞などの周辺情報を用いてブートストラップ的に評価表現を取得する方法として那須川ら [8] の研究がある。これはブログから最新の評価表現を取得可能である。しかし、Kamps らと同様に評価極性を二値で扱っており、それ以上詳細な極性、すなわち本研究でいう評価極性値を表すことはできない。

本稿で提案する評価表現辞書に類似した辞書として、熊本ら [9] は印象辞書を生成している。熊本らは R.Plutchik の提案した感情モデルに基づく「楽しい⇔悲しい」や「うれしい⇔怒り」などの 6 本の印象軸に対して類語辞典より文脈に影響されない 42 語の印象語を選別し、それらに印象値を付与する。印象値は新聞記事中での各印象語の共起関係から算出する。この印象値が本稿における辞書極性値に類似するが、我々は印象軸ではなく評価軸を用いる。また本稿の評価表現辞書は文脈依存の評価値や文脈理解の必要な評価値を扱うため、より多くの評価表現に対応できる。そして評価極性値を評価属性毎に出力する点が異なる。

2.3 評判情報の抽出と分類

評判情報の抽出には多くの場合で評価表現辞書が用いられ、評判情報の書かれた文書を肯定、または否定に分類することが多い。その方法としては評価極性の比率を用いる方法と機械学習による方法があり、前者には Turney の、後者には Pang ら [10] の研究がある。前者は肯定/否定表現の比率が高ければ文書を肯定/否定に分類する方法である。一方後者の機械学習では、サポートベクトルマシンなどを

利用し、肯定や否定の文書を学習することで分類する。

2.4 評判情報の可視化

評価属性の抽出と可視化に関連して Liu ら [11] の研究がある。Liu らは二値極性付きのレビューテキストを用いて評価属性を抽出し、肯定、否定の表現の割合を棒グラフで出力するシステムを提案している。この手法は評価表現そのものを本研究のように数値化するものではなく、レビューテキスト内の評価表現の頻度を可視化する。

また評価属性毎のスコア計算と可視化に関しては Scaffide ら [12] の研究がある。Scaffide らは Amazon*2 のレビューテキストとそれに付与された満足度を用いて、評価表現の点数を評価属性毎に出力するシステムを提案しており、評価表現辞書を使用しない点が本稿の提案と異なる。

3. 評判情報に含まれる評価要素

はじめに抽出対象とする評判情報の要素を以下の通り定義する。

- 評価者 … 評価を行った主体
- 評価対象 … 製品など特定のクラスの実体を指す表現
- 評価ドメイン … 評価対象の属するドメイン
- 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の値を示す表現（「良い」など）
 - 評価属性 … 評価対象の属性（側面）を示す表現（「音質」など）

本稿では、実験の対象データとして価格.com*3 のレビューを利用する。この価格.com のレビューではほとんどの場合、評価者と評価対象、評価ドメインは自明である。そこでレビューテキストから係り受け解析器 CaboCha-0.63*4 を用いて評価表現のみを抽出する。本研究では CaboCha の出力する“形容詞”と“形容動詞”を評価値の候補とし、“名詞”を評価属性の候補とする。なお、CaboCha の出力する各品詞を以下のように連結する。

- 名詞の連結
 - 連続する「名詞」
 - 「接頭詞-数接続」「名詞-数」
 - 「接頭詞-名詞接続」「名詞-一般」
 - 「形容詞-自立」「名詞-接尾-特殊-サ」
- 形容詞、形容動詞の連結
 - 「形容詞-自立」「動詞-非自立」「助動詞-ズ」
 - 「名詞-一般」「形容詞-自立」
 - 「名詞-一般」「名詞-接尾-形容動詞語幹」
 - 「名詞-サ変接続」「名詞-接尾-形容動詞語幹」
 - 「名詞-ナイ形容詞語幹」「助動詞-ナイ」

*2 <http://www.amazon.co.jp/>

*3 <http://kakaku.com/>

*4 <http://chasen.org/taku/software/cabocha/>

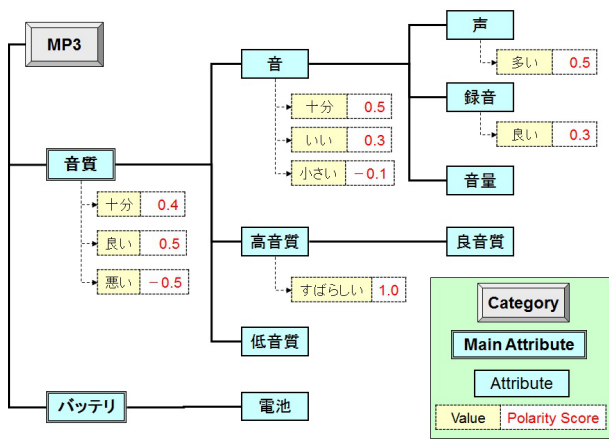


図 1 評価表現辞書の例

- 「名詞-ナイ形容詞語幹」「助詞-格助詞-一般」「形容詞-自立」
- 「形容詞-自立」「助動詞-ナイ」
- 「形容詞-自立」「助詞-係助詞」「助動詞-ナイ」
- 「形容詞-自立」「助詞-係助詞」「形容詞-ナイ」
- 「名詞-サ変接続」「動詞-自立」
- 「名詞-形容動詞語幹」「助動詞-ダ」「助動詞-ナイ」
- 「名詞-形容動詞語幹」「助動詞-ダ」「助詞-係助詞」「助動詞-ナイ」

4. 評価属性の関連付けと評価表現辞書の生成

4.1 評価表現辞書の生成

本研究で使用する評価表現辞書は、階層化された評価属性に評価値と辞書極性値を登録した辞書である(図1)。ここでは評価属性間の関連性を木構造で表している。また辞書極性値は[-1,1]の実数値であり、正の値が大きいほど肯定、負の値が大きいほど否定の度合いが大きいことを表す。

この評価表現辞書の生成手順を図2に示す。学習データとして価格.comのレビューテキストを用いる。レビューテキストより主属性に関連する評価属性を取得し、各主属性と関連付ける。ここでいう主属性とは図3に示した価格.comにある評価項目のことである。また同時に辞書二つ組を抽出する。辞書二つ組は辞書に登録するための「評価属性、評価値」のペアのことで、それぞれレビューテキストをCaboChaを用いて係り受け解析することで抽出する。さらに辞書極性値はレビューの評点を用いて計算する。この評点とは、価格.comにおいて評価主体であるレビューが評価対象に与える主属性ごとの点数のことである。

4.2 評価属性の階層化

評価属性の関連性を表す木構造を生成するアルゴリズムを図4に示す。始めに主属性を根の子ノードに割り当て、gen_mainAttribute_directryを用いて木構造を生成する。そして、gen_directoryで各主属性の子孫に割り当てられる評価属性が各主属性の関連属性である。

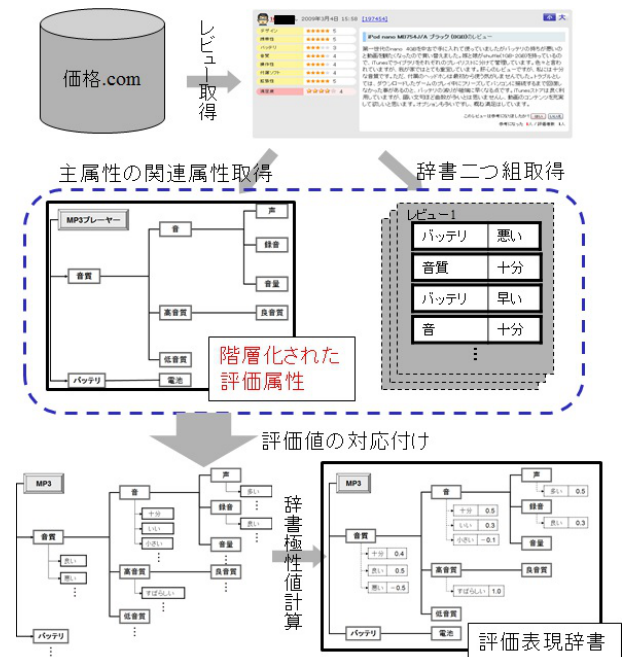


図 2 評価表現辞書の生成手順

評価項目	採票平均	カテゴリ平均	項目別ランキング	評価基準
デザイン	★★★★★ 4.69	4.42	14位	見た目のよさ、質感
携帯性	★★★★★ 4.59	4.44	23位	軽さ、コンパクトさ
バッテリー	★★★★★ 3.71	3.93	33位	バッテリーの持ちがよいか
音質	★★★★★ 3.62	4.20	37位	音質のよさ
操作性	★★★★★ 4.52	3.84	10位	操作性のよさ
付属ソフト	★★★★★ 4.00	3.39	13位	付属ソフトは使いやすいか
拡張性	★★★★★ 4.32	3.52	8位	対応オプション、アクセサリなどの充実度

図 3 ドメイン“mp3”における評価項目(主属性)と評価基準

図4の4行目のMainAttributesは主属性の集合であり、5行目のCandidatesが主属性の関連属性候補の集合である。関連属性候補の詳しい取得方法は4.3節で述べる。(c.attr).isValueBasis(n.attr)はnが主属性ノードの時に、関連属性候補cがnの関連性の評価基準に合致する場合にtrueを返す。関連性の評価基準は価格.comで用いられている基準を用い、図3はその例を示している。具体的には、図3の評価基準を形態素解析して得られる名詞の文字列と、関連属性候補cの文字列が完全に一致する場合、nとcは関連性があると定義する。また(c.attr).isRelated(n.attr)は関連属性候補cが特定の評価属性nの関連属性である場合にtrueを返す。この関連性の決定方法は4.3節で詳しく述べる。

4.3 関連評価属性の抽出

主属性の関連属性を抽出するために、レビューテキストより関連属性の候補を取得する。このとき、レビューテキストを適当に分割し、この分割単位のことをブロックと呼ぶ。関連属性の候補は以下のブロックに含まれる名詞全てとする。

```

1 void gen_mainAttribute_directory(){
2     depth = 2;
3     RootNode R;
4     MainAttributeNodes Main_Attributes;
5     AttributeNodes Candidates;
6     R.append_child_nodes(Main_Attributes);
7     gen_directory
8     (R.get_child_nodes(),Candidates,depth);
9 }
10 void gen_directory(N,C,depth){
11     if(depth > 0){
12         for(n : N)
13             for(c : C)
14                 if((n ∈ Main_Attributes &&
15                     (c.attr).isValueBasis(n.attr)) ||
16                     (c.attr).isRelatedTo(n.attr)){
17                     n.append_child_node(c);
18                     C.remove(c);
19                     break;
20                 }
21         for(n : N)
22             gen_directory
23             (n.get_child_nodes(), C, depth-1);
24     }
25 }

```

図 4 評価属性の木の生成

- 一文節
- 一文
- 箇条書き

一文節は CaboCha の出力するチャンク、一文は「。」で区切られたテキストとする。また図 5 に示すレビューの例の点線が示すような区切りを箇条書きブロックとする。箇条書きには図 5 に示す通り、「デザイン」に対する「背面」のような有効な関連属性を含む場合がある。また表 1 に 2011 年 10 月末から 2011 年 11 月末までの価格.com のレビューが箇条書きを含む割合を示す。表 1 より、過半数のレビューが箇条書きを含む記述になっていることが分かる。そこで箇条書きをブロックに加えた。なお、箇条書きは図 5 の丸で囲んだ主属性の出現を手掛かりにしてテキストを分割する。

次に評価属性間の関連性を決定する尺度として、TFIBF と平均情報量を定義した。評価属性の集合を $AttributeSet = \{a_1, \dots, a_n\}$ 、その関連属性候補を $TermSet = \{t_1, \dots, t_m\}$ とする。ここでは $AttributeSet$ を主属性の集合とし、その関連属性候補の集合を $TermSet$ とした場合を例に、それらの関連性を決定する尺度とした TFIBF と平均情報量について説明する。

4.3.1 TFIBF

TFIBF は情報検索でよく用いられる TFIDF の考え方にに基づき、主属性とその関連属性候補との共起頻度を文書 (Document) ではなく、ブロック (Block) を利用して以下

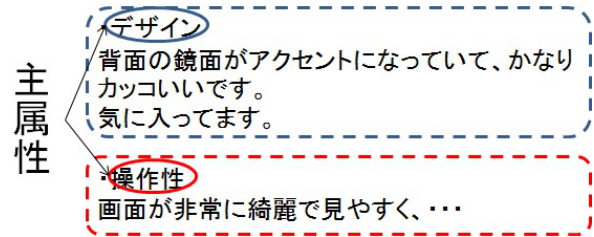


図 5 価格.com のレビューにおける箇条書きの例

表 1 箇条書きを含む価格.com のレビューの割合

ドメイン: パソコン > MP3 プレーヤー		
箇条書きを含む	箇条書きを含まない	合計
64	52	116

のように計算する。

$$TFIBF(a_j, t_i) = TF(a_j, t_i) \times IBF(t_i) \quad (1)$$

$$TF(a_j, t_i) = \frac{tf(a_j, t_i)}{N_{a_j}} \quad (2)$$

$$IBF(t_i) = \log \frac{N}{bf(t_i)} \quad (3)$$

ここで $tf(t_i, a_j)$ は、主属性 a_j と関連属性候補 t_i のブロック内での共起頻度、 N_{a_j} は主属性 a_j とブロック内で共起する全ての関連属性候補の共起頻度、 N は取得したブロックの総数、 $bf(t_i)$ は関連属性候補 t_i の出現するブロック数である。この $TFIBF(a_j, t_i)$ が大きいほど、関連属性候補 t_i の主属性 a_j との関連が強いとみなす。

4.3.2 平均情報量

主属性とその関連属性候補との共起頻度の偏りを計算したものが平均情報量となる。平均情報量は以下のように計算する。

$$Entropy(t_i) = - \sum_j P(a_j, t_i) \log P(a_j, t_i) \quad (4)$$

$$P(a_j, t_i) = \frac{tf(a_j, t_i)}{\sum_k tf(a_k, t_i)} \quad (5)$$

すなわち、 $Entropy(t_i)$ が大きいほど t_i はどの a_j とも共起しやすく、言い換えればいずれか特定の主属性との結び付きは強くない。そこで平均情報量が大きいものは関連属性から除く。

4.4 辞書二つ組の抽出

評価属性と評価値は図 6 に示す通り、一文中で互いに係り受けの関係になることが多い。図 6 は CaboCha による評価値の係り受け解析の例である。辞書二つ組の抽出では、はじめに評価値となる「形容詞」等を見つけ、次に以下から評価属性を探索する。

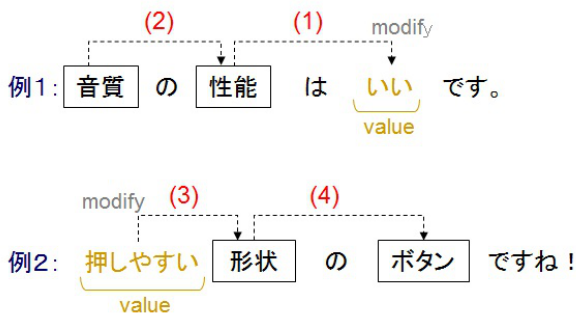


図 6 評価値の係り元と係り先

- (1) 評価値の係り元
- (2) 評価値の“二つ係り元”
- (3) 評価値の係り先
- (4) 評価値の“二つ係り先”

図 6 において、例 1 の「性能」が評価値の係り元であり、「音質」が“二つ係り元”である。同様に例 2 の「形状」が評価値の係り先であり、「ボタン」が“二つ係り先”である。例 1 において、評価値「いい」の係り元である「性能」は評価属性である。しかし、「性能」の係り元である「音質」も「いい」に対する評価属性である。よって評価値の係り元の係り元である“二つ係り元”も探索する。評価値の係り先についても同様に“二つ係り先”まで探索する。例 2 では、評価値「押しやすい」の係り先の「形状」とその係り先の「ボタン」を評価属性とする。

これら全ての評価属性と評価値とのペアを抽出し、評価表現辞書の対応する評価属性ノードに、辞書二つ組として配置する。

4.5 辞書極性値の計算

辞書極性値の計算には、価格.com のレビューテキストに付与された評点を用いる [13]。まずレビューの主属性毎の評点を、そのレビューテキストから抽出した辞書二つ組に付与する。この評点は {1, 2, 3, 4, 5} のいずれかである。そして評価属性と評価値が同じ辞書二つ組を収集し、その集合を *Pairs* とする。そして *Pairs* の評点の平均 *mean* を求め、値の範囲が [-1, 1] となるように正規化した式 (6) で定める *normal* をその二つ組の辞書極性値とする。

$$normal = \begin{cases} 2 \cdot \frac{mean - min}{max - min} - 1 & (if \ max > \ min) \\ 2 \cdot \frac{mean - 1}{4} - 1 & (if \ max = \ min) \end{cases} \quad (6)$$

ただし式 6) で *max* は *Pairs* の中で最大、*min* は最小の評点である

5. 評価表現の評価極性値の計算

4 節の評価表現辞書を用いて、評判情報の評価極性値を計算する。計算の概略を図 7 に示す。まずレビューテキストから「評価属性、評価値」の評価二つ組を抽出する。「評

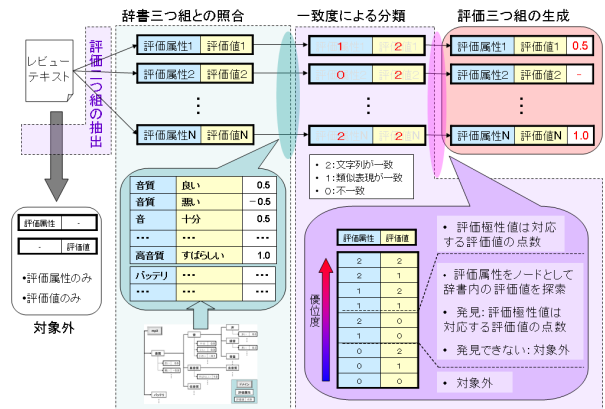


図 7 評価極性値計算の概略

価属性」と「評価値」の抽出方法は 4.4 節で説明した辞書二つ組の抽出と同様である。また「評価属性」のみ、あるいは「評価値」のみしか抽出できなかった場合は評価極性値計算の対象としない。これら評価二つ組を評価表現辞書と照合し、それぞれの評価極性値を 5.2 節に示す方法で計算することで評価三つ組を生成する。

5.1 評価表現辞書との一致度

可視化対象のレビューテキストから抽出した評価表現と、評価表現辞書のエントリの一致する度合いを一致度と呼び、評価属性と評価値のそれぞれの一致度を以下のように定める。

- 文字列が一致...2
- 部分文字列が一致...1
- 不一致...0

評価二つ組の評価極性値の計算はこの一致度を考慮して行う。以後、評価二つ組「評価属性、評価値」の辞書エントリとの一致度を (評価属性の一致度, 評価値の一致度) のように表記する。

5.2 評価表現の評価極性値の計算

5.1 節で説明した一致度により、次のように評価二つ組を分類する。

- (1) 評価属性と評価値がともに完全一致... (2,2)
- (2) 評価属性と評価値の少なくとも一方が部分一致... (2,1), (1,2), (1,1)
- (3) 評価属性のみ一致... (2,0), (1,0)
- (4) その他... (0,2), (0,1), (0,0)

上記 (1) (2) のグループの評価二つ組には評価表現辞書の対応するエントリの辞書極性値をそのまま用いて、評価三つ組「評価属性、評価値、評価極性値」を生成する。

(3) のグループでは評価表現辞書から評価属性とは関係なく対応する評価値を探索する。評価値が見つければ、評価極性値 *polarity* を式 (7) で計算する。

$$polarity = [\vec{v}, \vec{w}] \quad (7)$$

ここで \vec{v} と \vec{w} はそれぞれ式 (8), (9) のベクトルで, また式 (10) が成り立つ.

$$\vec{v} = [v_1, v_2, \dots, v_n] \quad (8)$$

$$\vec{w} = [w_1, w_2, \dots, w_n] \quad (9)$$

$$w_1 + w_2 + \dots + w_n = 1 \quad (10)$$

式 (8) の v_i は辞書で見つかった i 番目の評価値の辞書極性値, w_i はそれに対応する重み, n は探索で見つかった評価値の数である.

また (4) のその他のグループでは評価極性値の計算は行わず, 評価三つ組は生成しない.

6. 評価実験

TFIBF と平均情報量を用いて関連する評価属性を自動抽出し, その適合率について評価した. さらに, 5.2 節の方法で計算した評価極性値と, レビューテキストに付随する評点との相関係数を計算し, 評価極性値の妥当性について検証した. 評価極性値の妥当性評価では, 類語辞典 Weblio*5 と類似文字列を用いて作成した評価表現辞書 [14] を用いて計算した場合と比較した. ここで類似文字列とは文字列として包含関係にある評価属性のことである.

評価実験には, 価格.com のドメイン「MP3 プレーヤー」に含まれる, 2005 年 12 月から 2011 年 11 月までの 6 年分 7395 件のレビューテキストを使用した. そのため, この「MP3 プレーヤー」の主属性は図 3 に示した評価項目である.

6.1 主属性の関連属性抽出

6.1.1 TFIBF による関連属性の抽出

表 2 に TFIBF 値に基づいて抽出した主属性の関連属性上位 10 件とその主属性, TFIBF 値, および適合判定を載せる. 適合判定は図 3 の「評価基準」に基づき, 本稿の第一著者の判断で属性間の関連性が明らかな場合を rigid 判定で正解とし, 文脈によって関連性が認められる場合, もしくは他の主属性とも関連性が認められる場合を relaxed 判定で正解とした, 一方で明らかに関連がない場合は false とした. 例えば表 2 の「一文節」の場合, 2 位の「x アプリ」は SONY 社の提供している付属ソフトであり, 明らかに「付属ソフト」と関連性が認められるので rigid 判定で正解とした. 一方 4 位の「動画」は, バッテリーの持ちを「動画」の再生時間と対比させる場合があるため, relaxed 判定で正解とした. また「簡条書き」から抽出した 10 位の「私」はどの主属性とも関連を持たないと考えられるので false とした.

*5 <http://thesaurus.weblio.jp/>

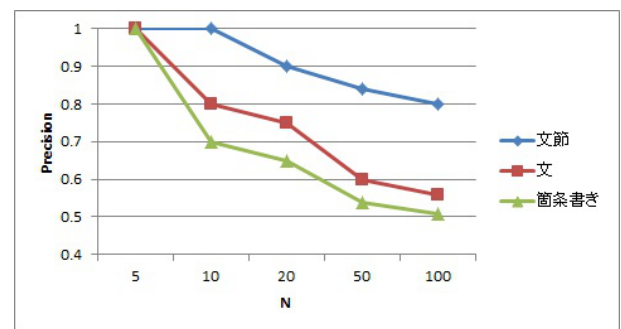


図 8 各ブロック単位 N の Precision@N

表 2 より, 一文節とする場合, TFIBF の高い関連属性には有用なものが多いことが分かる. 一方で, ブロックを一文, または簡条書きとした場合にはノイズが含まれている. これは抽出範囲が一文節に比べ大きいことが一因であると考えられる. また, ブロックが適切に区切られていない可能性もある. 例えば, 一文は「。」で区切られた場合としているが, 実際には「。」がない場合やその他の記号で区切られている場合がある. よって, より適切な粒度のブロックで分割を行うことで, 関連評価属性の抽出精度が上がる可能性がある.

図 8 に, 文節, 文, 簡条書きのブロックから TFIBF に基づいて抽出した属性の精度 Precision@N を示す. なお N は関連性を判定した関連属性候補の数 (上位 N 件) で, rigid または relaxed の関連性が認められるものを適合とした. この図より, 文節で区切った場合の適合率が最も高いことが分かる. よって, 6.2 節の実験で用いる評価表現辞書の生成には文節で区切った場合の TFIBF を用いた.

6.1.2 関連属性候補の平均情報量

表 3 に関連属性候補の平均情報量の上位 10 件と, その適合判定を載せる. この適合判定では, 図 3 の評価基準と照合し, 各評価属性候補がいずれか一つの主属性と関連が認められる場合を rigid 判定で正解とし, 文脈によって関連が認められる, もしくは複数の主属性と関連が認められる場合を relaxed 判定で正解とした. また, いずれの主属性とも関連が認められない場合を false とした.

表 3 の「一文節」の「今」や「前」などはどの主属性とも関連しない. また 1 位の「cowonj3」や, 7 位の「iphonephoto」などは製品名であるため false となる. この表 3 より, Entropy が高い関連属性候補はどの主属性とも関連が弱いことが分かる. ただし, 「簡条書き」の 6 位の「ソフトケース」は主属性の「携帯性」や「拡張性」と関連があると考えられる. また 8 位の「ホルダー」も「携帯性」や「拡張性」に関連がある. このように「簡条書き」では「一文節」や「一文」に比べ, 若干関連属性が認められる. このことは表 4 から分かる. 表 4 は平均情報量により各分割単位に基づいて抽出した関連属性候補の Precision@100 の値を表し, この値が小さいほど関連属性として不適切なもの

表 2 TFIBF で抽出した関連属性とその主属性

一文節				
Rank	関連属性	主属性	TFIBF	適合判定
1	アクセサリ	拡張性	0.325	rigid
2	x アプリ	付属ソフト	0.288	rigid
3	x-アプリ	付属ソフト	0.225	rigid
4	動画	バッテリー	0.141	relaxed
5	胸ポケット	携帯性	0.134	relaxed
6	サイズ	携帯性	0.130	rigid
7	microsd	拡張性	0.118	relaxed
8	通勤	バッテリー	0.112	relaxed
9	ブラック	デザイン	0.108	rigid
10	付属	付属ソフト	0.093	rigid

一文				
Rank	関連属性	主属性	TFIBF	適合判定
1	itunes	付属ソフト	0.205	rigid
2	ボタン	操作性	0.188	rigid
3	音	音質	0.135	rigid
4	x アプリ	付属ソフト	0.125	rigid
5	胸ポケット	携帯性	0.120	relaxed
6	ipod	拡張性	0.119	false
7	イヤホン	音質	0.095	relaxed
8	sonicstage	付属ソフト	0.084	rigid
9	ケース	拡張性	0.076	relaxed
10	十分	バッテリー	0.074	false

箇条書き				
Rank	関連属性	主属性	TFIBF	適合判定
1	音	音質	0.081	rigid
2	胸ポケット	携帯性	0.063	relaxed
3	itunes	付属ソフト	0.060	rigid
4	イヤホン	音質	0.049	relaxed
5	操作	操作性	0.044	rigid
6	ソフト	付属ソフト	0.039	rigid
7	ipod	音質	0.037	false
8	曲	付属ソフト	0.034	false
9	イコライザ	音質	0.033	rigid
10	私	デザイン	0.032	false

表 3 平均情報量の大きい関連属性候補

一文節			
Rank	関連属性候補	Entropy	適合判定
1	cowonj3	2.790	false
2	今	2.711	false
3	前	2.673	false
4	自分	2.670	false
5	4	2.624	false
6	本体	2.622	false
7	iphonephoto	2.585	false
7	a808	2.585	false
9	以前	2.582	false
10	不満	2.503	false

一文			
Rank	関連属性候補	Entropy	適合判定
1	機能	2.751	false
2	3g	2.725	false
2	雲泥の差	2.725	false
4	最大	2.721	false
5	満足度	2.714	false
5	何	2.714	false
7	僕	2.712	false
8	iphone	2.708	false
9	最近	2.706	false
10	あまり	2.705	false

箇条書き			
Rank	関連属性候補	Entropy	適合判定
1	5分	2.780	false
1	今日	2.780	false
3	一緒	2.778	false
4	心配	2.777	false
5	^^	2.772	false
6	ソフトケース	2.771	relaxed
7	存在	2.761	false
8	オークション	2.759	false
8	ホルダー	2.759	relaxed
8	usb メモリ	2.759	false

表 4 平均情報量による関連属性候補の Precision@100

	文節	文	箇条書き
rigid	0.02	0.01	0.22
rigid + relaxed	0.15	0.04	0.35

が多い。表 4 の「rigid」は rigid の関連性が認められる評価属性が含まれる場合、「rigid+relaxed」は rigid、または relaxed の関連性が認められる場合である。この表より、箇条書きを分割単位とした場合は他の二つに比べ、明らかに false の含有率が低いといえる。

図 9 に文節ブロックから取得した関連属性候補の TFIBF と平均情報量の散布図を載せる。これは文節ブロックから抽出した 2337 個の関連属性候補から無作為に 210 個取り出し、横軸をその TFIBF 値、縦軸を平均情報量として各候補をプロットしたものである。この図より、rigid な関連性が認められる評価属性候補は、false のそれに比べ、TFIBF 値が大きく、平均情報量が小さい傾向にあることが分かる。そこで評価表現辞書の生成の際は、TFIBF > 0.05、Entropy < 2.50 を満たす関連属性候補を用いた。

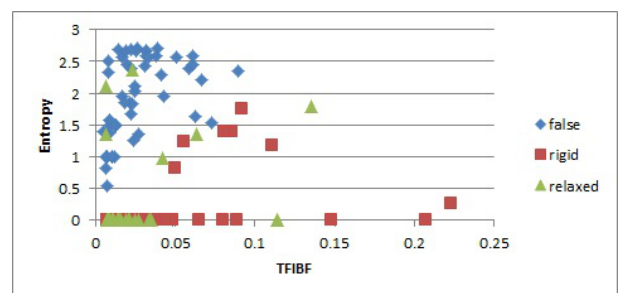


図 9 主属性の関連属性候補の散布図

6.2 評価極性値の妥当性評価

生成した評価表現辞書を用いて計算したレビューの評価極性値の妥当性を評価する。具体的には 7395 件のレビューテキストの内の 3 分の 1 を評価表現辞書の生成に用い、残りを評価表現の評価極性値の計算に用いる三分割交差検定を行い、評価極性値とレビューの評点との相関係数、ならびに可視化率を算出した。

ここで、可視化率は評価極性値を計算できた評価表現を含むレビューテキストの割合を主属性ごとに計算した値である。可視化率は評判情報の評価極性値への変換率を表し、1 に近いほど多くの評価表現の評価極性値が計算できたことを示す。また相関係数は 1 に近いほど、計算された評価極性値は妥当であると言える。なお評価極性値を式 (7) で計算する場合の重みは式 (11) のように定めた。

$$w_1 = w_2 = \dots = w_n = \frac{1}{n} \quad (11)$$

また、Weblio 類語辞典と類似文字列を用いて生成した評価表現辞書を利用した結果と比較した。

図 10 は評価極性値の主属性別の相関係数の平均で、横軸は主属性の種類、縦軸は交差検定の 3 回の実験における相関係数の平均である。この図より、本稿で提案した辞書生成法は Weblio と類似文字列を用いる場合に比べて相関が高いことが分かる。よって本稿の方法で生成した評価表現辞書を用いて計算した評価極性値には一定の有効性があると考えられる。

一方図 11 は評判情報の可視化率で、横軸は主属性、縦軸は可視化率を表す。この図では、拡張性と携帯性を除き、可視化率は本稿の提案の方が低かった。可視化率も大きいほどよいので、今後は本稿の提案と Weblio 類語辞典と類似文字列を併用して、関連属性を取得する方法などについて検討したい。

またレビューの中に比較表現が含まれる場合は、評点と評価極性値が逆になることがある。例えば特定の商品「A」について述べたレビューにおいて「A より B の方が良い」という一文がある。これは「A」については否定的な表現だが、本手法では「A」が「良い」とみなされる。対処方法としてはレビューの評価対象の同定を行い、比較表現を手掛かりとして正しい評価極性値に変換することが考えられる。またレビューテキスト内で肯定表現と否定表現の両方が存在する場合、評価極性値とレビューの評点が合致しない場合がある。一方で「形容詞」、「形容動詞」だけではなく、「動詞」や「助動詞」も評価値になりえるため、これらを利用することで可視化率の改善が期待できる。

7. まとめ

我々は評判情報を評価属性毎に数値化し可視化するシステムを提案している。本稿では我々が主属性と呼ぶ特定の評価属性の関連属性を、TFIBF 値と平均情報量を用いて

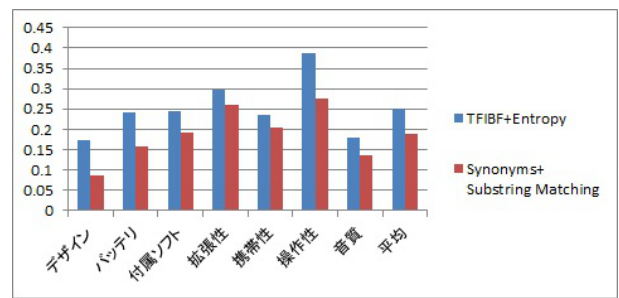


図 10 評価極性値と評点との相関係数

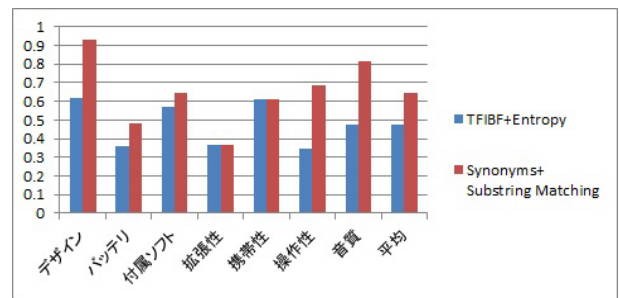


図 11 レビューテキストの可視化率

自動抽出し、それを評価表現辞書に利用する方法を提案した。評価実験では、抽出した関連属性の適合率を示し、文節を区切りとして抽出した関連属性の適合率が良いことを確認した。さらにこの評価表現辞書を用いて評判情報を主属性毎に評価極性値に変換し、その適切性を評価した。具体的には相関係数と可視化率を用いて評価し、Weblio 類語辞典と類似文字列を用いて評価表現辞書を作成した場合と比較した。

関連属性の抽出実験では、文節を区切りとして関連属性を抽出した場合の適合率が高かったが、文や箇条書きの区切りでのみ抽出可能な関連属性も存在する。そのため、テキストを文や箇条書きを区切りとして分割するアルゴリズムを見直したい。また本研究では木構造を用いて評価属性を主属性に関連付けているが、関連付けの方法や構造についてさらに詳しく検討する必要がある。今後は特に相関係数の改善を図り、評判情報可視化システムのプロトタイプを実装して被験者実験等を実施したい。

参考文献

- [1] 西田 京介, 藤村 考: 階層的オートタギングによる Q&A コミュニティの知識整理, The 2nd Forum on Data Engineering and Information Management (DEIM2010), D3-4, 2010.
- [2] A. Agresti: A Survey of Exact Inference for Contingency Tables, Statistical Science, vol.7, no.1, pp.131-153, 1992.
- [3] 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, The 2nd Forum on Data Engineering and Information Management (DEIM2010), A2-3, 2010.
- [4] 山田 敬之, 安村 禎明, 上原 邦昭: 各属性のレビュー・評価値の関係をを用いた評判情報の検索支援, 電子情報通信学会技術研究報告, 信学技報, vol.107, no.480, pp.1-6,

- 2008.
- [5] J. R. Quinlan: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Machine Learning, vol.16, no.3, pp.235-240, 1994.
 - [6] Jaap Kamps and Maarten Marx and Robert J. Mokken and Maarten de Rijke: Using WordNet to Measure Semantic Orientations of Adjectives, The 4th International Conference on Language Resources and Evaluation(LREC-2004), 2004.
 - [7] Turney Peter D: Thumbs up? Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews, The 40th Annual Meeting of the Association for Computational Linguistics(ACL-2002), pp.417-424, 2002.
 - [8] 那須川 哲哉, 金山 博: 文脈一貫性を利用した極性付評価表現の語彙獲得, 情報処理学会自然言語処理研究会(NL-162-16), pp.109-119, 2004.
 - [9] 熊本 忠彦, 河合 由起子, 田中 克己: 新聞記事を対象とするテキスト印象マイニング手法の設計と評価, 電子情報通信学会論文誌 (D), Vol.J94-D, No.3, pp.540-548, 2011.
 - [10] Bo Pang and Lillian Lee and Shivakumar Vaithyanathan: Thumbs up? Sentiment Classification using Machine Learning Techniques, The Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), pp.76-88, 2002.
 - [11] Bing Liu and Minqing Hu and Junsheng Cheng: Opinion Observer: Analyzing and Comparing Opinions on the Web, The 14th International World Wide Web Conference(WWW2005), 2005.
 - [12] Christopher Scaffidi and Kevin Bierhoff and Eric Chang and Mikhael Felker and Herman Ng and Chun Jin: Red Opal: Product-Feature Scoring from Reviews, Proceedings of the 8th ACM conference on Electronic commerce(FCRC-2007), 2007.
 - [13] 谷本 融紀, 太田 学: 評判情報可視化のための評価表現辞書の有効性評価, The 3rd Forum on Data Engineering and Information Management(DEIM2011), F2-6, 2011.
 - [14] 谷本 融紀, 太田 学: 評価表現辞書を用いた評判情報の極性値計算, The 4th Forum on Data Engineering and Information Management(DEIM2012), D2-3, 2012.