レビューからの商品比較表の自動生成

相川 直視† 山名 早人†‡

†早稲田大学大学院　‡国立情報学研究所

E-mail: {aikawa, yamana}@ yama.info.waseda.ac.jp

1. はじめに

意思決定をする際の判断材料として意見・評判情報は重要な役割を果たす. 例えば計画的に商品を買う際, 人気や価格帯により複数の購入候補を挙げ, その候補をより綿密に比較し一つを選ぶという工程を経る. Web 上のレビュー記事は, この比較の際の主要な情報源であるが, 全てのレビュー文を読むのには多くの時間がかかってしまう. これに対し, 一般に用いられている5段階評価の点数などでは, 数値に表れない定性的な評価や意見の対象の情報が欠落してしまう. そこで本研究では, 商品情報サイトの文章によるレビューを用い, 意見を適切な項目ごとにまとめ上げることで, 複数の候補を比較する表の作成を行った.

比較表を作成する際の主な問題は, 文章中から評価項目と評価値に相当する部分を抜き出すこと, 及び, 「価格」「値段」などの同じ内容ではあるが異なる表記を同一の項目としてまとめ上げることの 2 つである. 本稿では, 前者の問題を TF-IDF 値の高い名詞を評価項目, 形容詞を評価値のシードとして, 評価項目・値とその抽出ルールを交互に獲得していくブートストラップ手法により解決した. 後者の問題については, 評価項目とその内容を, 両者の関係性を考慮しつつそれぞれクラスタリングするアプローチをとった.

実験では, レビュー文から取得した比較表の項目と値がそれぞれ適切であるかの精度を測定するとともに, 人手で作成された評価項目がどれだけ再現できているかを測定した.

2. 関連研究

関連研究を, Web 上から, 表形式のデータを抽出する研究, 評判情報を抽出する研究の 2 つに分けて紹介する.

2.1. 表形式のデータ抽出

Web 上の文書から, 表形式で表されるような構造化されたデータを収集・構築しようという研究が存在する.

Arasu ら[1]は, Amazon.com などのデータベースを元にテンプレートにより生成されたページから, 元の DB スキーマを復元抽出するという研究を行った. Cafarella ら[2]は検索エンジン Google が収集した Web ページに含まれる大量の<table>タグで作成された表を解析し, 多くの表スキーマとそのデータ値を取得する研究を行った.

これらの研究は, 半構造データを, 元の構造化されたデータに復元しようという研究である. これに対し, 本研究は, 非構造データを入力としている点が異なる. 具体的には, レビュー記事という文章から, 表としてまとめられる部分のみを抽出し提示するというものである. このため, 文章中に多く存在する, 表の要素に入らない部分を除外しなければならないことと, それぞれの要素が項目なのか値なのかそのどちらでもないのか, どの項目とどの値が対応しているのかを判断しなければならない.

2.2. 評判情報抽出

意見評判情報を処理し, 自動的に意味のあるデータを研究しようという研究は多い[4][7]. ここではその中でも意見の評価対象, 評価項目, 評価値といったタプルを同定する研究に焦点を当てる.

Hu ら[3]は, 商品特徴とその意見の同定に関して, 頻度情報が大きな役割を果すことを示している. 小林ら[6]は, 評価の項目と値の対応付けでタスクの照応解析との類似性に着目し, 評価値をまず抽出し, 次に対応する評価項目を同定するという方法がより精度良く関係を対応付けることを示した. 評価項目と値の抽出には, これらが「<評価項目>は/が<評価値>」という形で出現しやすいことを利用したブートストラップによる手法で, 少ない手作業で作成した辞書を用いている.

これらの研究は, 評価項目・値の抽出自体か, その極性判断が目的であるのに対し, 本研究は評価項目・値の組を表にして提示するという具体的な応用目的がある. 表の要素は簡潔であることが望ましいため, 抽出の対象は単語や語句, 短文に絞られる. また, 複数商品の比較に使用するため, より多くの商品に共通して出現した評価項目のみ提示することが求められる. この結果, 出現頻度が多く, より確信度の高い評価項目・値のみを使用できるメリットが享受できる. 一方, 表現は異なるが意味が等しい評価項目をまとめる必要性が出る可能性がある.

3. 提案システム

3.1. 問題設定

本提案システムは，比較したい商品集合 $X = \{x_i | i = 1 \dots m\}$を入力として受け取る．候補商品集合 X に対し，X を比較するのに適した評価項目 $Y = \{y_j | j = 1 \dots n\}$と，その評価値 $V = \{v_{ij} | i = 1 \dots m, j = 1 \dots n\}$を求め，これらを図 1 に示すような表形式に表示ことを目標とする．

	商品名	y_1	…	y_j	…	y_n
評価対象X	x_1	v_{11}	…	v_{1j}	…	v_{1n}
	⋮	⋮	⋱			⋮
	x_i	v_{i1}		v_{ij}		v_{in}
	⋮	⋮		評価値V	⋱	⋮
	x_m	v_{m1}	…	v_{mj}	…	v_{mn}

評価項目Y

図 1 作成する比較表中の用語定義

3.2. 評価項目・値，その対応の獲得

1 つの商品に対してのレビューのみから，対応する評価項目と値を抽出するのはデータ数が少ないため困難である．そのため，まず比較候補商品全体のレビューから評価項目・値とその対応の DB を作成する．その後，作成された DB を用いて各商品に対しての評価項目・値を取得する．DB を作成する課題に対し，本システムでは，評価項目・値の DB と，それらを抽出するテンプレートを交互に獲得していくブートストラップ手法を用いる．具体的には以下の手順で処理を行う．

3.2.1. 候補単語の生成

一つ一つの単語を，対応関係を同定する評価項目・値の候補として扱う．まず，レビュー文を形態素解析器 MeCab により単語分割する．この際，「機能/性」などの連続する名詞，「非常/に」などの名詞-形容詞語幹+助詞-副詞化，「～っぽい，～やすい」などの接尾，非自立の形容詞とその直前の単語を 1 つの単語として結合した上で扱った．

3.2.2. シード DB の獲得

まず，初期シードとしてレビュー内の TF-IDF 値の高い数件の名詞を評価項目，形容詞・形容動詞語幹を評価値として 10 件ずつ DB に登録する．ここで TF-IDF 値は以下のように算出する．

TF(w) = レビュー中の w の出現頻度

DF(w) = Web 日本語 N グラム[1]上の w の頻度

$$\text{TF-IDF} = \begin{cases} \text{TF(w)} / \log(\text{DF(w)}) & (\text{DF(w)} > 1) \\ 0 & (\text{DF(w)} \leq 1) \end{cases}$$

[1] http://www.gsk.or.jp/catalog/GSK2007-C/catalog.html

直感的には Web 全体に比べてレビュー上で多く用いられる語句は評価語である可能性が高いことを利用している．

3.2.3. 抽出パターンの生成

次に，レビュー文中で近接した，DB 中の評価項目 y と値 v のペアを全て抽出する．この時 y, v 間の語数を s とし，s が 0 以上 3 以下であることを近接の定義とする．例えば，「機能性」と「良い」のペアからは，s が 0 では「良い/機能性」，s が 1 では「機能性/が/良い」，s が 2 では「機能性/も/とても/良い」などが抽出される．

この後，評価項目・値の一方を抜き取り，抜き取った箇所に入り得る全ての語句をレビュー中から探索する抽出パターンを作成する．例えば最初の例からは「<項目の候補>が良い」，「機能性が<値の候補>」の 2 つの抽出パターンが生成される．

3.2.4. 抽出パターンの抽出精度算出

例として，「機能性が<値の候補>」というパターンでの抽出結果を図 2 に示す．各抽出パターンによって実際に抽出された語句のうち，既存 DB に入っているもの，いないものの個数を T, F とする．各々，図中の白と灰色の部分に相当する．求めるべき抽出パターンの抽出精度 pは，実際にはカウントすることができない．

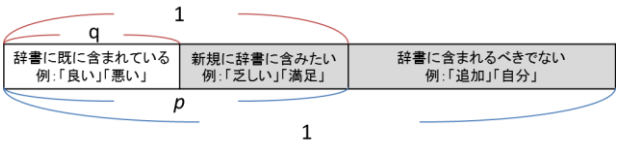

図 2 抽出パターンで抽出される語の分布

そこで，既に DB に入っている語数の，本来 DB に含まれているべき全語数からの割合 q を導入する事により本来の pを予測する．Q を用いると T は以下の式(1)で表すことができる．

$$T = (T + F) \times p \times q \quad (1)$$

これを予測抽出精度 pについて解けば，

$$p = \frac{T}{q \times (T + F)} \quad (2)$$

を得る．q を見積もり，式(2)に代入することで，予測抽出精度 pを算出できる． p が 1 を超えてしまった場合は，p =1 にする．q は今回のイテレーションで DB に含まれている個数を現在の x 倍にしたい場合，$q = 1/x$ として，見積もることができる．ただし，抽出パターンの良し悪しにより，抽出される個数は増減するので，スコアの高い順に x 倍になる個数分 DB 登録する方法よりも，柔軟であると言える．実験では，q=0.6 とした．また，全て既存 DB に含まれている語しか抽出しない抽出パタ

ーンは予測抽出精度 100%となってしまうが，これはデータが少ないため生じる．既に辞書に含まれるべきでない語は 1 度抽出されていると考え，F に 1 を足した状態で計算するという add-one スムージングを行った．

3.2.5. 評価項目・値の抽出

ある単語 w が実際に評価項目や値である確率 $p(\mathrm{w})$ は，w を抽出した抽出パターンの集合を E として，以下の式(3)で表される．

$$p(\mathrm{w}) = \frac{\prod_{\mathrm{i}\in\mathrm{E}} p_i}{\prod_{\mathrm{i}\in\mathrm{E}} p_i + \prod_{\mathrm{i}\in\mathrm{E}} (1 - p_i)} \quad (3)$$

ここで p_i は各抽出パターンの確信度を表す．この式を用いると，例えば w が確信度 60%の抽出パターンと，確信度 70%の抽出パターンに 1 度ずつ表れた場合，$0.6 \times 0.7 / 0.6 \times 0.7 + 0.4 \times 0.3$ の約 77.8%の確率で w が DB に登録されるべきであると算出される．0.5 以上のものを新たな評価項目・値として DB に追加する．

3.2.6. まとめ

3.2.3-3.2.5 の方法を繰り返す事で評価項目・値とその対応を獲得することができる．新しい評価項目・値や抽出パターンが抽出されなくなるか，イテレーションが規定回数に達したら終了する．繰り返しの最後で予測抽出精度が 0.5 以上となった抽出パターンが対応付ける評価項目・値の組を，対応する評価の項目と値の組とする．評価の項目と値が対応しているかは，イテレーションの最後で予測抽出精度が 0.5 以上となった抽出パターンによって結び付けられているかによって判定する．

この方法により完全に自動で評価項目と値の抽出ができるが，より精度が必要な場合は初期シードの入力や各イテレーションの DB 登録語の選定を人手行うことも考えられる．

3.3. 比較表の作成

3.2 で作成した評価項目・値の DB をもとに，各商品について評価項目と対応する評価値を抜き出すことにより，比較表が作成できる．この時，より多くの商品に共通する評価項目から順に抽出することで，誤って DB に登録された語の影響を軽減できる．

3.3.1. 同義評価項目のクラスタリング

「値段」「価格」といった意味は同じだが表記は違う項目に関しては，結合して表示した方がより望ましい．そこで，評価項目と評価値の対応関係など，2 部グラフに落とし込める関係を考慮してクラスタを作成できる Infinite Relational Model[5]を用い，クラスタリングを行った．

4. 評価実験

本手法の評価として，比較表要素として抽出した評価項目・値の精度を測定する．評価項目のクラスタリングの精度についてはあまりよい結果が出なかったため，良い結果が出なかった理由の議論をするに留める．なお，商品の比較は，カメラとパソコンといった別種のものでは行わず，同じ商品カテゴリ中の商品を比較するという仮定の元で実験を行った．

4.1. レビューデータの収集

実験に使用するレビューデータとして，価格.com[2]のレビューデータを用いた．データの収集は 2010 年 12 月 17 日時点で以下の手順にて行った．

1. 価格.com のカテゴリ詳細一覧ページ[3]から商品のカテゴリ名を取得する．
2. 価格.com の提供する「商品検索 API[4]」を利用し，1.で取得した各カテゴリ名で検索をかける．人気順に上位 20 件の結果を取得し，製品名，メーカー名，カテゴリ名，レビューページへの URL を使用する．
3. 検索ワードに入力したカテゴリ名と取得したカテゴリ名が異なるような結果を，収集対象から除外する．
4. API で取得したレビューページへアクセスした後，HTML 文を解析し，レビュー本文を取得する．
5. レビュー数が商品カテゴリ内の合計で 50 件以上のものを評価用データとして採用する．

この結果，123 品カテゴリから合計 23,969 件のレビューを集めることに成功した．3.の処理が必要になるのは，カテゴリ名で検索しても，そのカテゴリの商品が結果として返らないキーワードが存在するためである．

4.2. 評価項目・評価値抽出の精度評価

4.2.1. 実験方法

抽出した各表の要素の正確性と網羅性を定量的に評価したい．しかし，正解と言える比較表は，人手を用いても生成できないため，それらの測定は困難である．そこで，正確性の指標として，自動抽出された各評価項目・値についてそれらが適切である割合を用いた．網羅性の指標としては，評価項目のみに対して実験を行い，価格.com に存在する人手により予め定義された評価項目が再現できている割合を用いることにした．ここで，正確性の指標は適合率，網羅性の指標は網羅率と呼ぶことにする．適合率の評価は人手の作業を軽減するため，123 件の商品カテゴリ中からランダムに 25 件を選んで評価した．また，適合しているか微妙な場合は，適合数と非適合数に 0.5 ずつ加算した．

評価対象として，3.2.2 で述べたシード DB 自体と，

[2] http://kakaku.com/
[3] http://kakaku.com/sitemap/category_list.html
[4] http://apiblog.kakaku.com/

ブートストラップを行った後の抽出結果(表 1 中で+Bと記載)について, それぞれシードを作成する際に頻度順で抽出する方法と, TF-IDF により抽出する方法の 2 つを適用して実験した.

4.2.2. 実験結果

実験結果を表 1 に示す. 数値に表れない情報もあるため, シード DB の一例を表 2 に, ブートストラップで新たに抽出された評価項目・値の一例を表 3 に示す. 注意として, 表の上下が対応しているわけではない. 結果として, 評価項目の適合率, 網羅率は向上したが, 値の適合率は低下した. ブートストラップでは形容詞だけでなく名詞も抽出したことが原因としてあげられる. ただし, これにより,「当方所有」などの適切でない評価値が抽出された一方,「及第点」などの「良い」よりも良さの程度が分かりやすく, より有用な評価値が抽出されている.

4.3. 同義評価項目のクラスタリング

3.3.1 で述べたクラスタリングの結果を人手で見て評価した. その結果, 関係のないものが同一クラスタに属してしまう他, 全てが別クラスタに属してしまうなど, 良い結果が得られなかった. 人目で確認しても, そもそも 1 つにまとめたほうが良いと考えられる評価項目はほとんど存在していなかった. これは, レビューを書く際には既に書かれた他人のレビュー参考にするために同義の項目の用法はある程度統一が取れてしまっており, 使用頻度の低い評価項目はそもそも抽出されなかったと考えられる.

5. おわりに

本稿では, レビューを自動で比較表にして提示するシステムを提案した. 実際の利用の際には, 各評価値部分レビュー本文にリンクしておくといった UI 上の工夫が考えられる.

ブートストラップによる評価項目・値の抽出において, シード DB として Web 上の単語頻度を元にした TF-IDF を用いたこと, 各抽出パターンの予測抽出精度を定式化し, 現在の DB を大まかに何倍にしたいかという柔軟なパラメータを用いて抽出したことは, 評判情報抽出における本稿の新規点である.

今後の課題として, 同義の評価項目のクラスタリング, レビューの評価値を単語に限定せずに短文等も取得することが挙げられる. 後者については, 評価項目, 評価値の始まりと続き, その他, に相当する単語ラベル {Y, V_{start}, V_{cont}, O}を用意し, マルコフロジックネットワーク等によりラベル割り当てルールを学習していくブートストラップ手法[8]を用いれば実現する可能性がある.

表 1 作成した比較表の精度評価

精度指標	TF	TFIDF	TF+B	TFIDF+B
評価項目.適合	0.48	0.55	**0.57**	**0.62**
評価値.適合	**0.86**	**0.86**	0.63	0.66
評価項目.網羅	0.29	0.28	**0.41**	**0.44**

表 2 TF-IDF により抽出された初期シード上位 10 件の例(商品:デジタルフォトフレーム)

評価項目	画質	写真	フォトフレーム	デザイン	購入	リモコン	総評	画像	再生	プレゼント
評価値	良い	便利	高い	悪い	必要	十分	残念	キレイ	無い	綺麗

表 3 ブートストラップにより新たに抽出された評価項目・値の例

評価項目	色	縦横判別	スライドショー	数秒表示	何	写真枚数	店頭デモ	メモリー	自動電源 ON/OF
評価値	小さい	良好	及第点	最新	自動	かわいい	当方所有	色濃い	タイムマシン表示

参考文献

[1] A. Arasu and H. G. Molina, "Extracting Structured Data from Web Pages" SIGMOD, pp.337-348, 2003.

[2] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu and Y. Zhang, "WebTables: Exploring the Power of Tables on the Web" VLDB Endowment, pp.538-549, 2008.

[3] H. Hu and B. Liu, "Mining and Summarizing Customer Reviews" SIGKDD, pp168-177, 2004.

[4] 乾孝司, 奥村学, "テキストを対象とした評判情報の分析に関する研究動向" 自然言語処理, Vol.13, Num.3, pp.201-241, 2006.

[5] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada and N. Ueda, "Learning Systems of Concepts with an Infinite Relational Model", AAAI, 381-388, 2006.

[6] 小林のぞみ, 乾健太郎, 松本裕治, "意見情報抽出のための評価対象・評価視点間の関係同定" 言語処理学会第 12 回年次大会論文集, pp.65-68, 2006.

[7] B. Liu, "Sentiment Analysis and Subjectivity" Handbook of NLP, Second Edition, 2010.

[8] J. Zhu, Z. Nie, X. Liu, B. Zhang and J. R. Wen, "StatSnowball: a Statistical Approach to Extracting Entity Relationships" WWW, pp.101-110, 2009.