DEIM Forum 2013 D10-2

個人に対する有用性の高い Web ユーザレビューの推薦
係り受け解析器の比較検討および一文分割手法の検討

河中 照平† 井上 潮‡

‡東京電機大学工学研究科 〒120-8551 東京都 足立区千住旭町 5 番

E-mail: †12kmc13@ms.dendai.ac.jp, ‡inoue@c.dendai.ac.jp

あらまし 既存のインターネット通販サイトの多くは、他の閲覧者の評価の高いレビューを推薦している。しかし、レビューの用途は人それぞれであるため、閲覧者が求めていないレビューが推薦されてしまう場合があり、閲覧者自らが多くのレビューを読む必要があるため、それが閲覧者の負担になっている。本研究では閲覧者個人が求めるレビューを推薦することで、レビュー閲覧者にかかる負担を減らすため、閲覧者が必要な情報が含まれているレビューと判断したレビューを全て保存し、保存したレビューから評価語極性辞書と係り受け解析を用いて、閲覧者一人一人に対してユーザモデルを構築する。このユーザモデルに則り、閲覧者が求めるレビューを推薦することで、閲覧者にかかる負担を軽減する。

キーワード 情報推薦,自然言語解析

1. はじめに

商品に対するユーザレビューは様々な利用者の立場からの意見・評価・感想が得られるため、企業が出す情報では得られない有用な情報を得ることができる。一方でユーザレビューは多様であるため、質の良いものと悪いものが混在している。レビューの多さと信頼性の低さから、閲覧者がレビューの有用性を判断する必要があり、それが閲覧者の負担になっている。

Ｗｅｂユーザレビューは１つの商品に対して様々な立場、環境、状況の消費者、利用者がレビューを書くため、様々な視点が存在し、従来のレビューの問題点が克服されている。一方でレビューの書き手が一般人になったことで、レビューの信頼性は従来よりも高くない上に、読み辛いレビュー文を掲載する場合も多く存在する。また、書き手の人数が増えたことによって、レビューの数が膨大になる場合がある。

既存のインターネット通販サイトの多くは、他の閲覧者の評価の高いレビューを優先的に表示する、サイト側が規定した各評価項目でソートする、といった手法を取っている。しかし、レビューの用途は人それぞれであるため、閲覧者個人が求めている情報と照らし合わせたレビューを優先的に表示、推薦をしない限り、閲覧者が求めていないレビューが優先的に表示されてしまう。そのため、閲覧者自らが多くのレビューを読む必要があり、それが閲覧者の負担になっている。そこで、本研究では閲覧者の求めるレビューを推薦することで、レビュー閲覧者にかかる負担を減らすことを目的とする。

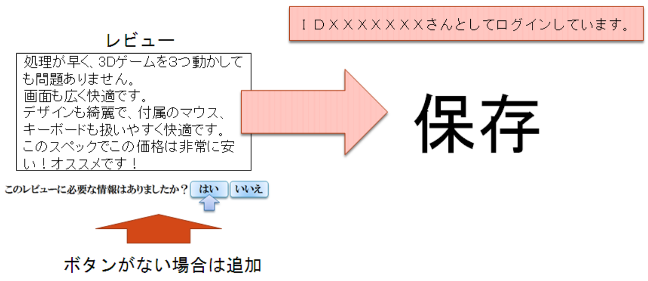
2. 関連研

小林ら[1]はレビューに対して自然言語処理を施し、単語の並びからタグ情報を文節に付加し、タグ情報の並びから、レビュー自体に有用な情報が含まれているかどうかを判定させる。これにより、有用な情報が含まれていないレビューの判別が容易になり、閲覧者へかかる負担を減らすことができる。しかしこの方法は、閲覧者に必要な情報が手に入るとは限らないという問題を抱えている。

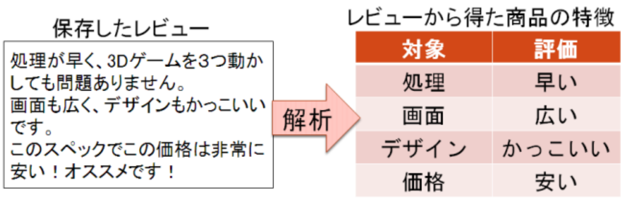
Hu ら[2]は商品のレビューからレビュアーが表明している意見、レビュアーの意見が正または負であるかどうか、レビューされている製品の特徴を要約する。これによって製品の特徴の要約から、どういった商品であるか容易に判断でき、閲覧者への負担を減らすことができる。本研究ではレビュー対象の特徴情報の抽出方法の手法を検討するにあたってこの研究を参考にした。

3. 提案手法

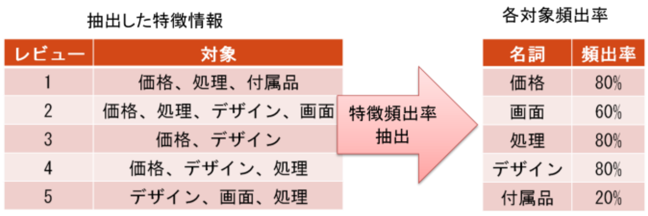
閲覧者個人が求めている情報を推薦するため、閲覧者が必要な情報が含まれているレビューと判断したレビューを閲覧者ごとに全て保存する（図 1-a）。保存した後、保存したレビューから評価語極性辞書[3]と係り受け解析を用いて、レビュー対象の特徴情報を抽出する（図 1-b）。そして抽出したデータを用いて、閲覧者一人一人に対してユーザモデルを構築する（図 1-c）。このユーザモデルに則り、閲覧者が求めるレビューを推薦することで、閲覧者にかかる負担を軽減する（図 1-d）。



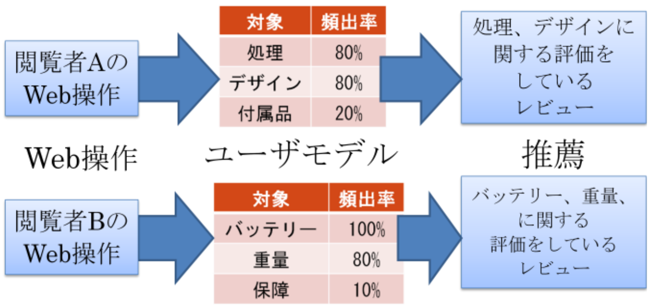
(a)



(b)



(c)



(d)

図 1.提案手法による処理の流れ

3.1. ユーザモデルの構築

ユーザの求めている情報に適応して動作するシステムを開発するうえでは、そのユーザの求めている情報を明示的にコンピュータに表現する。この表現をユーザモデルという。本研究では、このユーザモデルの構築に際し、レビュー閲覧者が必要な情報が含まれていると判定したレビューを全て保存し、その保存したレビューから情報を抽出し、ユーザモデルを構築する。なお、ユーザモデルの構築にはレビュー対象の特徴情報から構築し、そのデータベーススキーマで示したモデルを図 2 に示す。

図 2.ユーザモデルスキーマ

3.2. レビュー対象の特徴情報の抽出方法

レビュー対象の特徴的な情報が閲覧者の求める情報に大きく関わっていると考えられるため、本研究では、レビュアーが批評する内容はレビュー対象の特徴的な部分であると、Kim らの研究[4]から仮定した。そのため、乾らが作成した日本語評価極性辞書[3]を用いて評価語(特定の単語に評価を与える語句)を規定し、評価語が係り受けている語句を評価対象の候補として抽出することで、Web ユーザレビュー内からレビュー対象の特徴情報を抽出する。係り受けている語句の判定は係り受け解析が行う。

3.3. 係り受け解析器

一般に、係り受け解析器は受け取った文全てを一文とみなすため、一文に区切ってから係り受け解析に通さないと、二つ以上の文からなる文章を入れた場合には、はじめの文章の最後の単語が本来係り受けていない単語に係り受けていると判定されてしまう。Ｗｅｂユーザレビューはすべてのレビュアーが決まった構文でレビューを書いてはいないため、単純に句点だけで区切っただけでは一文として正しく区切ることができない。そのため、適当な個所で一文に区切ってから係り受け解析器に通す必要があり、一文として正しく区切る手法を検討する必要がある。そのため本研究では、正しい一文を係り受け解析器に送るために、自動でレビューを適当な箇所で分割するプログラムを作成した。

3.4. レビューの一文単位への分割方法

また、作成した一文分割システムは句点、2 段の改行、エクスクラメーションマーク、クエスチョンマーク、これらを仮切断箇所として、図 3 のフローチャートの流れにそって作成した。

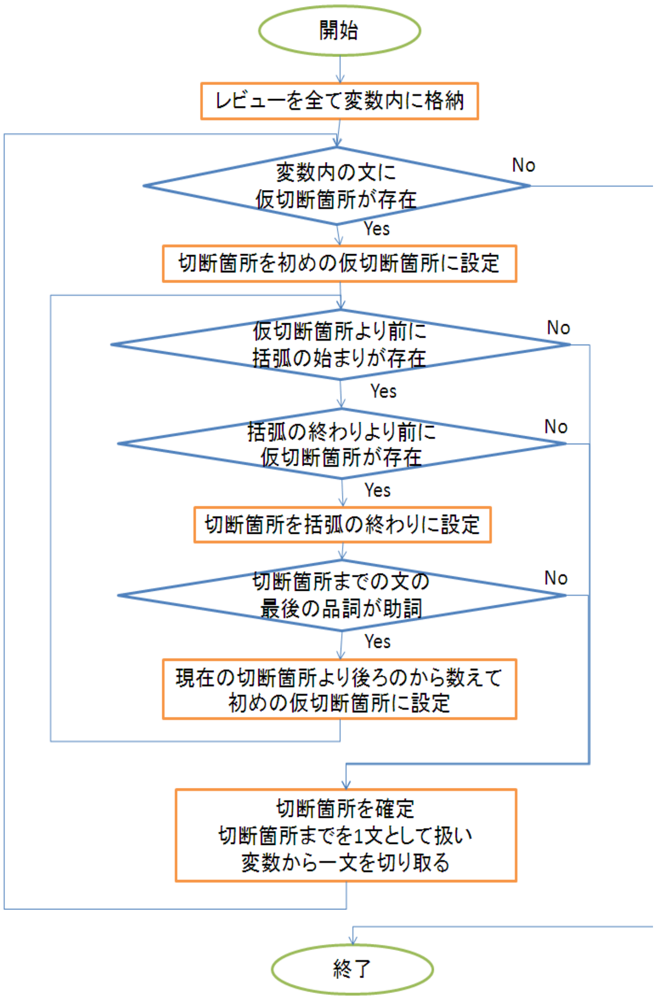


図 3.作成したシステムのフローチャート

4. 評価

係り受け解析器は、受け取った文を形態素解析し、その結果から文節を構築し、係り受け先の推定を行う。そのため、形態素解析が正しく行われない場合、係り受け解析の推定にも大きな影響を与えてしまう。

変換ミスや送り仮名の振り間違いなどが多く存在する Web ユーザレビューでは、形態素解析の精度が大きく変動すると考えられるため、本研究では使用する係り受け解析に用いられる形態素解析器の比較実験と、作成した一文分割システムの評価を行った。

4.1. 評価方法

形態素解析の比較には、辞書の内部が見ることができない Yahoo 日本語係り受け解析 API[5]と Cabocha の係り受け解析部分を担う Mecab [6](0.98pre3)に、最も本研究に適した形態素解析を行う IPA 辞書による形態素解析の結果を比較した。本研究では、この評価には 2012 年 5 月 29 日 16 時頃、「パソコン・周辺機器」において Amazon ランキング 1 位であった「EPSON インクカートリッジ(6 色セット) IC6CL50」の最新レビュー100 件に対して実行した。また、本研究で特徴情報の抽出の際に重要になる、名詞、形容詞を抜き取りその結果の比較を行った。

自作したシステムの評価には 2012 年 10 月 25 日、「パソコン・周辺機器」において Amazon ランキング 1 位であった Transcend SDHC カード 16GB Class10 永久保証 [フラストレーションフリーパッケージ (FFP)] TS16GSDHC10E のレビュー最新 100 件に対して行った。なお、この評価実験は一文の判定が正しく行われているかどうかが評価項目なので、一文しかないレビューは評価実験の対象から除外した。また、同様に日本語のレビューを対象としているため、英語のレビューも対象から除外した。これらのレビューを除外した上で最新 100 件に対して実行した。

4.2. 係り受け解析器の比較実験結果

形容詞、名詞における Yahoo 形態素解析、Mecab の計測数と計測した種類数の比較を表 1 に示す。この結果から Yahoo 形態素解析では形容詞が Mecab に比べ多めに分類され、Mecab は Yahoo 形態素解析に比べ名詞が多めに分類されていることがわかる。表 2 に示す形態素解析結果の名詞比較を見ると、Web ユーザレビューによく見られる略語を、YahooAPI は 1 つの単語として認識している。一方で Mecab は不良品を良品と不に分けているため、可能な限り細かく結果を返している。次に分解された形容詞の比較を行った。表 3 に Yahoo、Mecab どちらか片方にのみ分類された形容詞とその個数を、表 4 に形態素解析に失敗したと思われる結果を示す。

表 1. 各形態素解析の形容詞、名詞の計測結果

	形容詞	
	YahooAPI	Mecab
計測形容詞数	236	221
計測形容詞種類数	70	60
	名詞	
	YahooAPI	Mecab
計測名詞数	1540	1919
計測名詞種類数	629	659

表 2. 形態素解析結果の名詞比較

YahooAPI	Mecab
コスパ	コス__パ
不良品	不__良品

表 3. 各形態素解析の形容詞の分類結果

YahooAPI		Mecab	
片側分類単語	個数	片側分類単語	個数
問題なし	1	やすく	1
問題なく	2	っぽい	1
問題ない	2	すごく	1
正しく	1	かたい	1
仕方なく	1		
仕方ない	1		
巧く	1		
よろしく	1		
すばらしい	1		
しがたい	1		
お高い	2		
お安い	2		
い	1		
あまりよく	1		

表 4. 失敗したと思われる結果

YahooAPI		Mecab	
形容詞	い	名詞	づまりなどもありましたが

表 3 の Yahoo 形態素解析はお高い、お安いという単語を形容詞に分類し、巧く、しがたいといったあまり使われない単語も判別しており、必要以上に分解していないことがわかる。次に分解された名詞の比較を行った。その結果、Mecab の分解には記号の一部が名詞として扱われ、Yahoo 形態素解析では名詞として扱われる記号が存在しないという結果を得た。また、Yahoo 形態素解析において 1 文字のみで名詞に分解されたものが 7 種だけだったのに対し、Mecab では 45 種となった。このことから、Mecab は Yahoo 形態素解析と比べ細かく分類しているという結果が得られた。また、Web ユーザレビューを二つの形態素解析器に通した結果、失敗したと思われる結果が Yahoo 形態素解析、Mecab 共に見受けられた。YahooAPI の方は 1 文字の形容詞が判定されたが、Mecab 側は 12 文字の名詞が結果として表れた。Mecab の結果は明らかに形態素解析が不十分なので、係り受け解析器の文節の構築に問題を引き起こす可能性が高い。この結果から、Web ユーザレビューに対して Mecab は係り受け解析の結果を大きく左右する形態素解析の結果が得られたため、本研究では Mecab ではなく Yahoo 係り受け解析 API を使用することにした。

4.3. 作成したシステムの評価実験結果

100 件のうち 83 件が正しく一文ごとに区切ることができた。成功例を図 3 に示す。正しく区切ることのできなかった文の特徴としては、句点を打つべきところに読点を打っているレビュー、句読点をまったく打っていないレビュー、全角スペースごとに一文を区切っているレビュー、エクスクラメーションマーク、クエスチョンマークで一文が区切れていないレビュー、顔文字で区切っているレビュー等が挙げられる。この失敗例の分布を図 5 に示す。

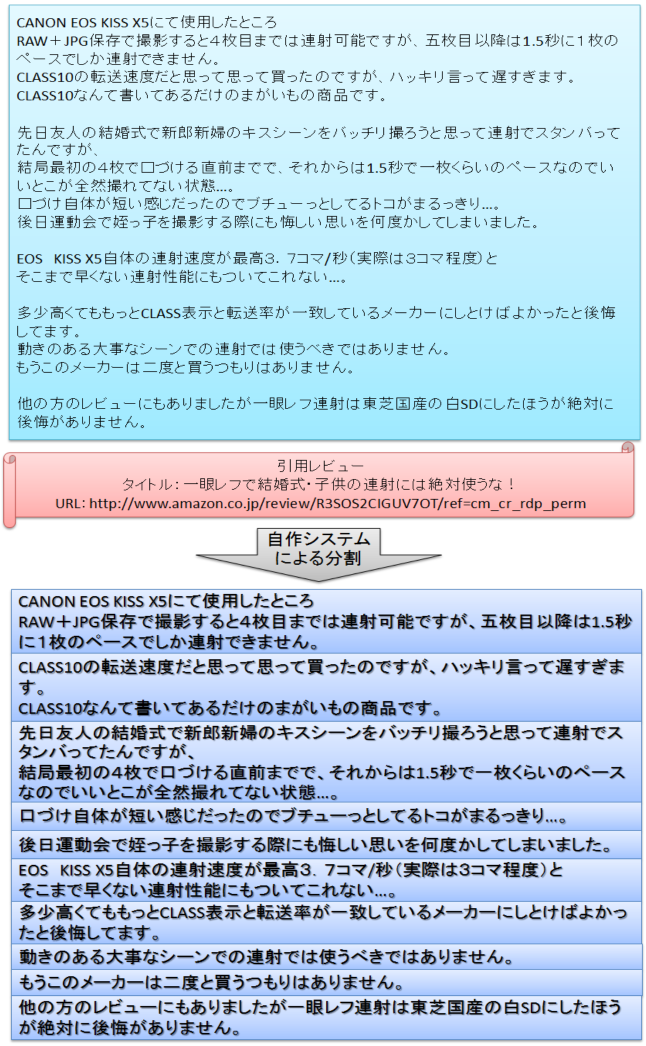


図 4.レビューの一文分割成功例

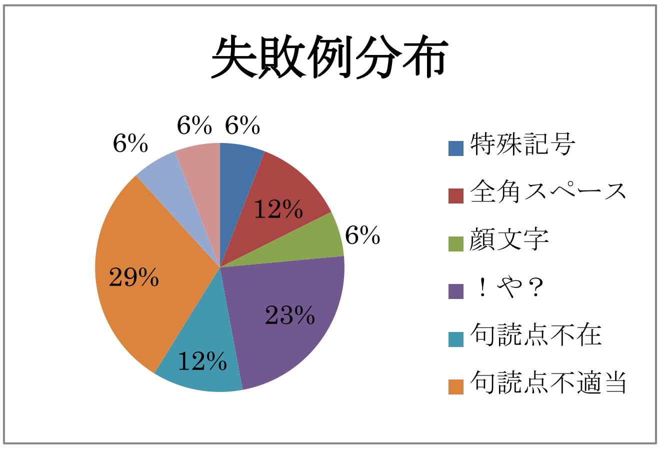


図 5 失敗例分布グラフ

5. 終わりに

本研究で用いる係り受け解析器の比較検討実験および、一文分割手法の検討を行った。

係り受け解析器の比較検討では、解析器に通す文章が Web ユーザレビューであるため、自然言語であることを考慮し、細かく切り過ぎず、かつ新しく作られた単語にも対応している Yahoo 係り受け解析 API を使用するものとした。

また、一文分割手法の検討では、作成したシステムが 80%を超える十分な精度を示したため、今後は、レビューから特徴情報を抽出するプログラムを作成した後、ユーザモデル構築システムを作成し、評価を行う。

参　考　文　献

[1] 小林大祐,井上潮,“ユーザレビュー情報における感情表現の分析方法の研究”,DEIM 2010 ,March 2010.

[2] Minqing Hu and Bing Liu ,“Mining and Summarizing Customer Reviews”, Department of Computer Science University of Illinois at Chicago,851 South Morgan Street,Chicago, IL 60607-7053

[3] 乾健太郎, 他 ″日本語評価極性辞書,″ http://cl.naist.jp/~inui/research/EM/sentiment-lexicon.html.

[4] S.M. Kim, P. Pantel, T. Chklovski, M. Pennacchiotti, ″Automatically Assessing ReviewHelpfulness,″ Proc. of the EMNLP 2006, pp.423-430, July 2006.

[5] Yahoo 日本語係り受け解析 API, http://developer.yahoo.co.jp/webapi/jlp/da/v1/parse.html

[6] Mecab,http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html