

# アスペクトごとの文の重要度と類似性判断に基づく複数レビューの要約

唯野良介 嶋田和孝 遠藤 勉

九州工業大学大学院情報工学府情報科学専攻

{r\_tadano, shimada, endo}@pluto.ai.kyutech.ac.jp

## 1 はじめに

近年、レビューサイトや Weblog などの普及に伴い、個人が意見やレビューを投稿する機会が増えている。これらの評価情報は有益であると考えられるが、Web 上には大量の情報が混在する。そのため、ユーザが望む評価情報を自動的に抽出・要約するシステムの必要性が高まっている。そのシステムの一つとして、我々はマルチアスペクトレビュー要約に着目する。対象が持つアスペクト(カメラの“画質”や“操作性”など)別に要約することで、偏りのない情報提示が可能となる。実際にマルチアスペクトレビュー要約を行う場合には、評価文抽出、アスペクト推定、要約生成などの複数のタスクが存在するが、本研究では複数レビューを対象とした要約タスクに焦点を当て、要約のための重要文抽出を目的とする。

この際に考慮しなければならない問題は、文の重要度の定義、重要文の抽出方法、冗長性の排除である。我々はこれらの問題に対応するため、要約文を決定する際の特徴要素としてレビューの評点、*tf-idf* 値、言及の多さの3つに着目する。言及の多さの測定と冗長性の排除には類似文クラスタリングを行う。そして、これらの結果を統合利用することで、より適切な要約の生成を目指す。

## 2 要約文抽出に用いる特徴

本研究で扱うレビューは、あらかじめ複数の評価視点(アスペクト)が与えられており、これに対してレビューアーが各アスペクトの評点(5段階評価など)とコメントを記述する形式とする。ある対象に対して異なるユーザのレビューが存在するため、要約対象は複数のレビューとなる。また、評価文の特定や評価文とアスペクトの対応付けは完了済みと仮定する。

本手法ではレビュー内から重要な評価文のみを選定し、要約文として提示する。要約文を抽出する際の判断要素として、次の3つの特徴を用いる。

- レビューの評点  
各アスペクトに付与された点数を特徴とする。
- *tf-idf* 値  
ある対象に固有な単語ほど重要とする。
- 言及の多さ  
言及が多い意見ほど重要とする。

上記3つの特徴要素をそれぞれ算出した後にこれらを統合利用することで、より適切な要約を生成する(図1)。以下、それぞれの特徴について説明する。

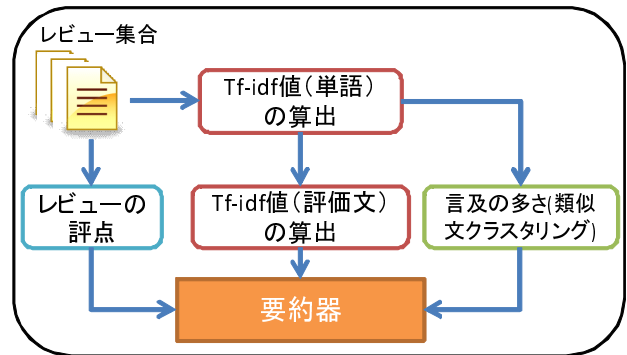


図1 手法の全体像。

### 2.1 評点

要約を生成する場合、要約対象全体の意見をバランス良く反映させる必要がある。例えば、肯定的な意見ばかり抽出してしまうと対象の否定的な面を把握することができず、読み手に正確な情報を伝えることができない。この問題を解消するための要素として、レビューに付与されている評点に着目する。レビューにはアスペクトごとにあらかじめ評点が与えられていることを利用し、レビュー内で出現した各評価文に評点を割り当てる。評点は、評価文に付与されたアスペクトに対応した点数となる。作成した評価文と評点のペアは、最終的に要約をする段階で評点の分布を考慮する際に用いる。

### 2.2 *tf-idf* 値

*tf-idf* 法は重要文抽出手法として頻繁に使われており、対象文書にのみ頻出する語を重要とする特徴を持っている。我々もこの *tf-idf* 法を導入し、評価文の重要度を算出する。まず、レビューに含まれるテキストを形態素解析<sup>\*1</sup>した後、単語の *tf-idf* 値を次式で算出する。算出対象は、名詞(非自立語、接尾語、代名詞、数を除く)と形容詞のみとする。

$$tf_j^i = \frac{\log_2(\text{対象レビュー } j \text{ での単語 } i \text{ の出現回数} + 1)}{\log_2(\text{対象レビュー } j \text{ での単語の異なり数})}$$

$$idf_i = \log_2\left(\frac{\text{全レビュー数}}{\text{単語 } i \text{ を含むレビュー数}}\right) + 1$$

最終的に、ある対象レビュー  $j$  内に出現する単語  $i$  の *tf-idf* 値は  $tf_j^i \times idf_i$  となる。次に、この値を用いて評価文の重要度を求める。ある評価文  $s$  の重要度は次のよ

\*1 形態素解析器には MeCab を用いた。  
<http://mecab.sourceforge.net/>

うに定義する。

$$\frac{\text{文 } s \text{ 中の単語の } tfidf \text{ 値の和}}{\text{文 } s \text{ 中で } tfidf \text{ 値をもつ単語数}}$$

この結果, 特徴的な語に着目した文の重要度が得られる。

### 2.3 言及の多さ

本研究では複数レビューを対象としているため, 異なるレビュアーによって同じ内容の意見が記述されることがある。このような同意見は, 要約文を抽出する際に重複し, 冗長性を生む可能性がある。一方で, 多くのレビュアーに言及される意見はその対象にとって重要な意見ともいえる。そこで, 冗長性を排除すると同時に言及の多さを測るために類似文の統合を行う。

#### 2.3.1 類似文クラスタリング

本手法では, クラスタリングを行うことで類似文の統合を図る。クラスタリングアルゴリズムには様々な種類があるが, 今回はクラスタリング手法として広く用いられている *k-means* 法を採用する。*k-means* 法は非階層的な手法であり, はじめに分割クラスタ数を指定する必要がある。しかし, 最適な分割クラスタ数をあらかじめ把握することは難しい。この問題に対して関ら [1] は, 各クラスタ数での結果を統計的に評価することで, 妥当なクラスタ数を推定している。これらの特徴から, 本手法においても関らの用いたアルゴリズムを利用する。

クラスタリングに用いるベクトル空間については, 各評価文を形態素解析し, 得られた形態素を素性としたベクトル空間を構成する。素性とする対象は名詞 (非自立語, 接尾語, 代名詞, 数を除く), 形容詞, 動詞のみとする。各素性の値は, 2.2 節で *tf-idf* 値が得られている語にはその値を与え, それ以外には評価文内での単純な頻度を与える。また, 素性をさらに特徴付けるために石井ら [2] が用いた単語の中心性という概念を導入する。これはテキスト中の話題格や主格, 目的格といった単語は話題の中心性が高いとし, それらの語に対して重み付けを行う手法である。本手法においてもこの概念を利用し, 評価文内の中心性が高い単語 (名詞) には重み付けを行う。これらのアルゴリズムを用いてクラスタリングを行うことで, 複数の類似文クラスタが形成される。

#### 2.3.2 クラスタリング結果の補正

2.3.1 節のクラスタリングアルゴリズムによって, 本タスクのデータをクラスタリングしたところ, クラスタが過分割される傾向があることがわかった。過分割が生じると, 同じクラスタに属すべき類似文が別のクラスタに属してしまい冗長性が残ってしまう。この問題に対応するため, クラスタリング結果に対してクラスタの統合・補正を行う。

上述したアルゴリズムでは, 各類似文クラスタは何らかの単語の共起によって集合した評価文で形成される。そのため, 各クラスタ内には共通して出現する単語が存在する。この特徴から, 各クラスタを代表する単語 (代表語) を特定することができる。本手法では各クラスタの代表語を比較することで, クラスタリング結果の補正を図る。代表語の条件は, 単一文で構成されたクラスタ

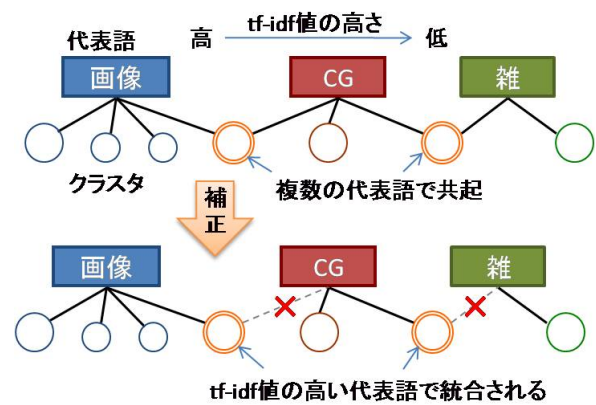


図2 クラスタの統合例。

であるか, 複数文で構成されたクラスタであるかによって次のように定義する。

- 単一文クラスタにおける代表語  
文を構成する全単語 (名詞, 形容詞)。
- 複数文クラスタにおける代表語  
文数/2 より多く共通して出現する単語。

代表語の利用方法として, 今回は代表語が一致するクラスタ同士を統合する。一つのクラスタに複数の代表語が存在する可能性もあるため, 統合相手となるクラスタが一つに定まらない場合がある。この場合は, 代表語の *tf-idf* 値を比較し, 高い方の代表語で一致するクラスタを優先する。クラスタの統合例を図2に示す。

## 3 要約器の構築

2 節ではレビューから要約文を抽出するための特徴要素としてレビューの評点, *tf-idf* 値, 言及の多さの3つを挙げ, その算出手法を述べた。本節ではこれらの特徴を統合利用し, 要約を生成する方法について述べる。

まず, 各文の *tf-idf* 値と言及の多さ (クラスタリング結果) を組み合わせ, 各クラスタの総合的な重要度を求める。あるクラスタ  $C_i$  の重要度を  $Imp(C_i)$  としたとき, その重要度を次の式で定義する。

$$Imp(C_i) = Ave_{tfidf}(C_i) \times \log(|C_i| + 1)$$

$Ave_{tfidf}(C_i)$  はクラスタ  $C_i$  に属する文の *tf-idf* 値の平均値であり,  $|C_i|$  は  $C_i$  に属する文の数である。

次に, レビューの評点を考慮した要約文の抽出を行う。要約文の抽出手順を次に示す。

1. 各クラスタから代表文を抽出する。今回はクラスタ内で最も *tf-idf* 値が高い文を代表とする。
2. 代表文が属する評点別に, 代表文を分類する。
3. 各評点に属する代表文数の割合を考慮し, 評点ごとに  $Imp(C_i)$  の高い文から要約文として抽出する。

上記の手順によって, 最終的に生成される要約文が決定する。

表 1 各状態におけるクラスタ数 .

アスペクト	適応前	適応後	補正後
オリジナリティ	110	45	24
グラフィックス	25	8	8
音楽	18	11	6
熱中度	47	30	20
満足度	84	47	26
快適さ	80	51	27
難易度	47	18	13

#### 4 実験

提案手法を実際のレビューに適応し、有効性を確かめた。実験データには、Shimada ら [3] が評価文書分類で使ったゲームレビュー文書を用いた。この評価文書には、各レビュー記事ごとに7つの評価項目が含まれている。評価項目は「オリジナリティ (o)」、「グラフィックス (g)」、「音楽 (m)」、「熱中度 (a)」、「満足感 (s)」、「快適さ (c)」、「難易度 (d)」である。評価文書には49本のゲームに対するレビュー記事が含まれており、全体のレビュー数は4174レビューである。この中から今回はNintendo DSソフト「New スーパーマリオブラザーズ」を実験対象とした。対象ゲームには170のレビュー記事が存在しているが、実際に要約対象となるレビューは任意に抽出した25レビュー(約450文)とした\*2。また、要約対象にはあらかじめ、アノータによって評価文の特定とアスペクトとの対応付けが行われている\*3。

##### 4.1 類似文クラスタリングの適応

まず、類似文クラスタリングの有効性を評価する。対象ゲームから抽出された評価文をアスペクトごとに分類し、各アスペクトに対してそれぞれ2.3節の手法を適応した。初期クラスタには  $tf-idf$  値の最も高い文と最も低い文の2文を与えた。クラスタリングによる各アスペクトのクラスタ数の変化を表1に示す。また、統合された文の例を図3に示す。表1の「適応前」は各アスペクトに分類された評価文数(1文を1クラスタとみなす)、「適応後」は2.3.1節のクラスタリングを適応した後のクラスタ数、「補正後」は2.3.2節の手法で補正をかけた後のクラスタ数を示す。

類似文の統合を行った結果、補正後のクラスタ数が適応前の状態に比べて平均で約3割まで削減された。適応後から補正後の変化に関してもクラスタ数の削減が確認でき、クラスタリング結果の補正は有効であるといえる。類似文の統合が上手くいった例としては、「快適さ」において「セーブ」という単語を含む記述が15文存在したが、補正後ではその内の14文が同じクラスタに含まれた。この場合「セーブ」は意見の主題となる場合が多いため、クラスタ内の文は全て類似した(記述の詳しさなどの差はある)内容となっていた。一方、「マリオ」という単語で共起することで、多くの文が同じクラスタへ統合されるという傾向がみられた。これは「マリオ」は今回の実験対象の特性として出現しやすいものであるため重要度( $tf-idf$  値)が高く、優先されたのが原因と考える。しかしこの場合、「マリオ」という単語は必ずしも意見の主題とはなっていない。そのため、意見の内容としては類似していないにも関わらず、必要以上に統合されているクラスタも存在した。この結果から、要約対象の特性として全体的に出現しやすい単語を特定し、それらの単語は重要度を下げるといった枠組みを考える必要がある。また、現在のアルゴリズムでは文の極性を

<ul style="list-style-type: none"> <li>1. セーブシステムが不便</li> <li>2. セーブが不自由なのも厄介だった</li> <li>3. セーブ地点がきまつてるのはよくない</li> <li>4. 自由にセーブができない</li> </ul>	<ul style="list-style-type: none"> <li>1. ロード時間も、殆どなく非常に快適</li> <li>2. 無駄に広い割に時間が短い</li> </ul>
<ul style="list-style-type: none"> <li>1. 難しいテクニックは必要なく、マリオを操作して面白</li> <li>2. 「マリオVSルイージ」の土管を壊して土管の位置にいと埋まってしまう</li> <li>3. 「とりあえずマリオするか」という感じで遊べ、プレイにしんどさを感じさせない</li> </ul>	<ul style="list-style-type: none"> <li>1. アクションは簡単でも爽快感がある</li> <li>2. すごく爽快感を削いでいる</li> </ul>
	<ul style="list-style-type: none"> <li>1. わかりやすい操作</li> <li>2. アクションの数が豊富な割には、操作方法が簡単</li> <li>3. 操作性に難あり</li> </ul>

図 3 補正後の結果の例.

考慮できないため、各クラスタ内で異なる極性を持つ文が存在する。これらについては素性に評点の情報を組み込むことや、各クラスタ内でさらに追加処理を行うなどして対応する必要がある。

##### 4.2 要約の生成

次に、提案手法を用いて実際にレビュー要約を行った。例として、「熱中度」の補正後の状態(20クラスタ)から10文を要約として抽出した結果を表2に示す。代表文とその  $Imp(C_i)$ 、評点に加えて、所属するクラスタとの関係性を調べるために、代表文が属するクラスタ内の評点の平均値「評点平均」と標準偏差「評点標準偏差」を共に示している。また、評点ごとの補正後の代表文数と実際に要約文として抽出された代表文数の対応関係を表3に示す。

表2の結果における問題点の一つは、評点と意見の内容が一致しない例が存在することである。例えば「コイン集めなどにはまった」や「一度クリアしてしまうと、もうやりたくなくなる」という要約文は、意見の内容と評点が矛盾する。これらは、対応するレビューの評点が全体的に低めであったり、意見の一貫性がないのが原因であった。そのため、レビュアーによって評点のゆらぎが生じることも考慮して評点を扱う必要があるといえる。他にも、代表文の評点と所属クラスタ内の評点平均に差があり、クラスタ内で評点にばらつきがある例も確認された。この原因としては、クラスタリングをする際に意見の極性(評点)は考慮できていないことや、クラスタ内の意見の類似度が低い場合が考えられる。また、クラスタから抽出した代表文を評点ごとに分類する際に、代表文の評点のみを参照したことも原因である。そのため、より正確な文の統合に加えて、代表文の選定方法も改善しなければならない。

一方、表3の結果から、評点のバランスを考慮した重要文抽出に関しては適切な結果が得られたといえる。

\*2  $tf-idf$  値の算出には170レビュー全てを用いている。

\*3 本実験では著者が行った。

表2 「熱中度」における要約文抽出結果.

要約文として抽出された代表文	$Imp(C_i)$	評点	評点平均	評点標準偏差
操作性も悪くて死んでばかりでストレスがたまり、すぐ飽きました.	2.64	0	0.00	0.00
コイン集めなどにはまった.	8.35	2	3.40	1.43
何か物足りないゲームだった.	4.01	2	4.00	1.00
隠しステージやクリア後のオマケが無い.	4.28	3	3.33	1.25
なんだか進めていても単調に思える時がある.	1.77	3	3.00	0.00
中毒になっちゃう.	1.56	3	3.00	0.00
巨大マリオでステージを破壊しまくる爽快感.	6.30	4	4.60	0.49
隠しステージやメダル集めといったやりこみ要素があり、... あまり飽きを感じなかった.	4.25	4	2.50	1.66
一度クリアしてしまうと、もうやりたくなる.	3.94	4	2.00	1.22
新しい発見があったり、こんなテクニックもあるのか、と気づいたりして何度でも楽しめる.	1.61	5	5.00	0.00

表3 「熱中度」の評点ごとの代表文数の分布.

評点	0点	1点	2点	3点	4点	5点
補正後の文数	1	0	5	5	7	2
抽出された文数	1	0	2	3	3	1

しかし予備実験において、レビューの評点の性質として(対象の質にもよるが)、平均的な意味を示す3点や4点が与えられやすいことが確認された。そのため現在の手法では、どのレビューにおいても3点や4点に属する要約文が多い結果が予想される。この場合、0点や5点などの割合が減ることで対象の特徴を把握できなくなる可能性もあるため、より有効な各評点からの抽出方法を考える必要がある。

## 5 関連研究

レビュー要約を対象とした先行研究として、Mengら[4]は製品の仕様・特徴から必要なアスペクトを抽出し、階層構造を用いた要約を行っている。その結果、製品ごとに適切なアスペクトの抽出を可能としている。しかし、生成された要約は製品に対する具体的な意見を含んでおらず、要約として十分な情報量とはいえない。それに対して提案手法では、特徴的な語を含む重要文を抽出することでより詳細な意見の把握が可能である。

Blair-Goldensohnら[5]はWordNetを用いて算出した単語の極性値(p/n)とレビューの評点を利用し、最大エントロピー法を用いて文の極性値を算出している。そして、要約文として極性値の高い評価文を優先して抽出している。この手法の利点として、各文の極性を高い精度で推定できる点が挙げられる。しかし、極性値の高い評価文が要約文として必ずしも重要であるとは限らない。また、要約文の内容の重複も考慮されていない。

一方、Luら[6]はPLSAにアスペクトの概念を導入し、専門家によるレビューとWeb上に散在する意見の統合を行っている。統合する意見は、Web上に存在する意見の数から話題の大きさを測ることで、重要な意見を選定している。この手法は有効であるが、Luらの目的は1つのレビューを基に、それに対する類似文や補助文を付加させることで情報量を増やすことである。そのため、多くの情報から重要な部分を選定する我々のタスクとは厳密には異なる。

## 6 おわりに

本稿ではマルチアスペクトレビュー要約を目的とし、レビューの評点、 $tf-idf$ 値、言及の多さに着目した要約作成手法を提案した。言及の多さの測定と冗長性の排除として、クラスタリングを用いて類似文の統合を行った。クラスタリングについては実験の結果、対象文を約3割まで削減することができた。また、実際に要約を生成した結果、文の重要度と評点のバランスを考慮した要約文の抽出をすることができた。今後は、類似文の統合手法、評点の扱い方、代表文の選定方法などの改善を図り、より良い要約生成を目指す。また、有効性の評価方法として、人手による要約を用いた定量的な評価も行いたいと考えている。

## 謝辞

この研究の一部は栢森情報科学振興財団の助成を受けて遂行された。

## 参考文献

- [1] 関恒仁, 嶋田和孝, 遠藤勉. 表の属性と属性値の関係を利用した類義語抽出. 電子情報通信学会論文誌, Vol. J89-D, No. 9, 2006.
- [2] 石井弘志, 林日華, 古郡廷治. 単語の中心性に基づくテキスト自動要約システム. 情報処理学会研究報告, 2001.
- [3] K. Shimada and T. Endo. Seeing several stars: a rating inference task for a document containing several evaluation criteria. In *Proceedings of PAKDD 2008*, 2008.
- [4] Xinfan Meng and Houfeng Wang. Mining user reviews: from specification to summarization. In *Proceedings of ACL-IJCNLP 2009*, 2009.
- [5] Sasha Blair-Goldensohn et al. Building a sentiment summarizer for local service reviews. In *Proceedings of WWW 2008: NLPix Workshop*, 2008.
- [6] Yue Lu and Chengxiang Zhai. Opinion integration through semi-supervised topic. In *Proceedings of WWW 2008*, 2008.