

書評利用のレコメンデーションを目的とする 書評検索システムのためのキーワード抽出

越本 和季 酒井 浩之 増山 繁
豊橋技術科学大学

1 はじめに

ネット上の小説投稿サイトや小説検索サイトには、多数の小説が登録されている。そこでは、評価基準として、主にランキングが用いられている。しかしながら、ランキングは値の集計による評価基準であるため、そこから分かるのは多数の人に好まれているか否かである。したがって、自分好みの小説を探すことには直接には利用できず、そのような小説を探すのは手間がかかる。そこで、個人の好み合ったレコメンデーションシステム作成を目標とする。

本研究は、提案するレコメンデーションシステムの一部である書評検索システムの作成を目的とする。現段階ではネット小説のみを対象としている。ネット小説を対象としたのは、大量の書評を容易に集めることができる点と、書評とその対象小説を対応付けることが容易である点が、提案レコメンデーションシステムに向いているためである。また、iPadやKindleの発売にみられるように、電子書籍の普及が予測される。一般書籍も電子化されることにより、ネット小説と同様に、読むことと、書評を投稿することを同一の端末から行うことが可能となる。このことから、今後、電子化された書籍に対する書評の増加が予想される。これらの書評をその対象小説と対応付けることができれば、提案レコメンデーションシステムを用いることができる可能性がある。

以下、ネット上の小説投稿サイトや小説検索サイトにおけるレコメンデーションについては2章で、提案システムについては3章で、書評検索のためのキーワード抽出法を提案手法として4章で、評価実験については5章で、考察を6章で述べ、7章を統括とする。

2 ネット小説に対し、現在運用されているシステムで用いられているレコメンデーション機能

実際に小説投稿サイトや、小説検索サイトで用いられているレコメンデーション機能として、以下のものがある。

● ランキング^{1, 2, 3}

アクセス数や、感想数、話数などによりソーティングしたものである。サイト独自のポイント(評価値)を用いたランキングも存在する。感想投稿時に5段階で評価させることによるポイントや、一定期間に一度行われる面白いかな否かの投票によるポイントなどが用いられている。

● 検索、口コミ掲示板^{4, 5}

条件を入力することで、条件にあった小説を紹介してもらうことができる。多くの人が訪れるサイトの掲示板ならば、非常に強力である。他のレコメンデーションと異なり、そのサイトに登録されていない小説も紹介される。ただし、利用は自分ですできるだけ探す努力をした後であることが前提である。

● ピックアップ^{1, 4}

条件に合った小説の中から、ランダムに数作品を選んで推薦する。条件は、ジャンル指定や一定以上人気があることなどである。

● お気に入り登録機能¹

お気に入り登録機能を利用したレコメンデーションである。ユーザは好きな小説をお気に入り登録することができ、他のユーザのお気に入りリストを確認できる。また、各小説に対して、その小説をお気に入り登録している人たちが他にどの小説をお気に入り登録しているかを確認できる。ログイン制を前提とした機能であるため、用いられているサイトは少ない。

また、レコメンデーション機能ではないが、小説を探す上で有効な機能として以下のような検索機能がある。

● 属性キーワード検索^{1, 2, 3, 4}

ユーザに提示される、属性キーワードを用いた検索機能であり、選択した属性キーワードに属する小説を検索することができる。属性キーワードは、サイトへ小説の登録をする際に作者側が小説の属性として相応しいものを設定する(属性キーワードの具体例は4.2節の表1で示している)。サイト側が予め用意した選択肢の中から設定する場合と、作者が任意の語を設定する場合がある。ランキングと併用することで、ランキングにおいては個人の好みは考慮されないという欠点のある程度補うことができる。例えば、SF小説のみを対象としたランキング等が可能である。ランキングが存在せず、属性キーワード検索のみがある場合もある。

● 任意キーワード検索^{1, 2, 3, 4, 5}

ユーザは任意の語を入力することで、それを題名や作者名などに含む小説を検索できる。読みたい小説の題名や作者名が、予め分かっている場合に有効である。

¹HinaProject, 小説家になろう, <http://syosetu.com/>

²ネット小説ランキング, <http://nnr.netnovel.org/>

³NEWVEL, <http://www.newvel.jp/>

⁴ChaosParadise, <http://chaosparadise.jp/>

⁵Arcadia, <http://www.mai-net.net/>

これらの機能は、すべて小説検索であり、書評に対しては対応していない。本研究では、書評の検索機能は小説のレコメンデーションに利用することが可能であると考え、これの作成を目指すこととした。

3 個人の好みに合ったレコメンデーションのためのシステムの提案

提案システムは、個人の好みに合ったレコメンデーションを目的としたシステムである。提案システムのレコメンデーションは書評を利用する。本研究では、小説を読んで感じたこと、考えたことを記す文章を書評と定義した。本稿で紹介するのは、レコメンデーションシステムの書評検索システム部分となる。提案するレコメンデーションシステム全体の概要を 3.1 節で、提案する書評検索システムの概要を 3.2 節で、これらのシステムの流れを図式化したものを図 1 に示す。すなわち、書評検索システムと、推薦までの流れである。

3.1 提案するレコメンデーションシステム

書評には、書評者がその小説(小説 A とする)のどこが好きかといった情報が含まれている。その好みは、当該ユーザと書評者で似通っている場合、書評者とは小説全体に対する好みも似通っていると判断する。好みの似た書評者が、面白いと評価した小説 B は、当該ユーザにとっても面白いと仮定する。また、小説 A は当該ユーザにとって既読である必要がある。

ユーザが共感できる書評か否かの判断はユーザ自身に書評を読んでもらうことで行う。提案の書評検索システムはユーザが気に入る書評を探しだすことの支援を目的としたものである。

書評検索システムは、2 節で述べた既存のシステムには存在しない。一方、共感できる書評者を探し出した後の処理は、お気に入り登録機能を持つサイトならば代用できる。

3.2 提案する書評検索システム

書評検索は、書評検索用のキーワードを書評内から抽出して実装する。ここで書評検索キーワード(以下、単にキーワード)は小説内の人物名や地名、「設定」「世界観」といったような語である。ユーザは与えられたキーワードから興味を持ったものを単数、もしくは、複数選択することで、それらを含む書評を検索できる。このキーワード群は特性として以下を持つことが望ましい。

- 全ての具体性を持つ書評が検索可能であることが望ましい

ユーザが読む価値のある書評であるか否かの判断基準として、書評者の、小説に対する具体的な感想が記述されているような書評(具体性を持つ書評)を、読む価値のある書評とする。例えば、「面白い」「つまらない」といった評価のみが記述されている書評には具体性はなく、それらの評価に対する理由が記述されている書評が具体性のある書評となる。

- 意味が明白であることが望ましい
キーワードから小説内のシーンが連想可能であれば、意味が明白。

- 意見でないことが望ましい
検索に用いるキーワードとしては、意見でない方が良い。例えば、あるシーンに対する書評として複数の異なる意見がある場合、検索単位が、意見単位でなく、シーン単位であれば、ユーザは興味のあるシーンに対する多様な意見を読むことができる。

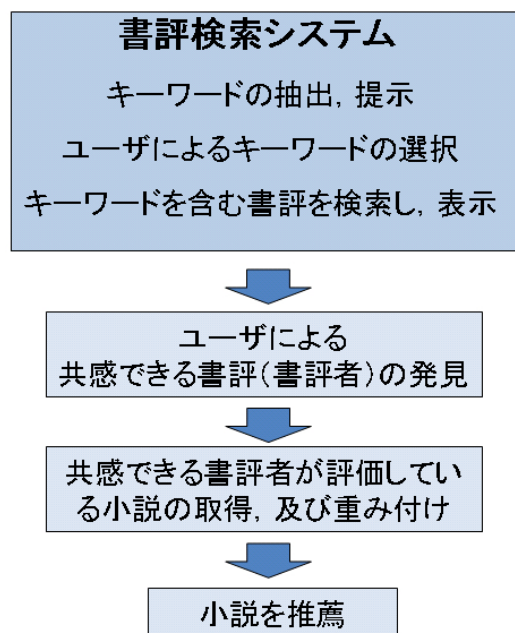


図 1: 書評検索システムと、推薦までの流れ

4 提案手法

書評検索に用いるキーワードの抽出法を提案する。基準となるキーワード抽出法を 4.1 節に示す。本手法は、異なる小説の書評間でのキーワードの出現頻度の差を利用している。本節では、小説の本文と、それに対する書評を組としたものを指して「小説」としている。4.2 節の処理は、出現頻度の差に、ジャンルによる偏りが発生しないように行う処理である。4.3 節の処理は、出現頻度の差では抽出できない語を抽出するための処理である。

4.1 キーワード抽出法

使用するコーパスは、対象小説の書評コーパス(対象小説コーパス)と、対象小説以外の書評を集めたコーパス(フィルタ用コーパス)の 2 種類である。フィルタ用コーパスは、対象小説コーパスから、キーワードとして相応しくないものを除去するためのコーパスである。キーワードとして相応しくないものとは、「楽しみ」「感想」といったものである。これらは、どのような内容の書評にも含まれている可能性があり、書評を検索するためのキーワードとして相応しくない。また、キーワードの候補語(以下、候補語)は、名詞と

複合名詞とする．小説 N_i を対象小説として，その書評内に存在する候補語 w を以下の式により重み付けする．重み付け後，降順でソートする．キーワード数の上限を設定し，上位からその数だけキーワードとする．候補語 w に対して，(1) 式による重みは， w が書評内に出現する小説が，フィルタ用コーパス内に少ないほど大きくなり，対象小説コーパスに w の出現回数が多いほど大きくなる．そのため，最も重みが大きくなる w の条件は，対象小説の書評にのみよく出現することとなる．

$$Score_i(w) = \log(1 + df_i(w)) - \log\left(\frac{n_{filter}}{df_{filter}(w)}\right) \quad (1)$$

$df_i(w)$: 対象小説コーパスにおいて， w が含まれている書評数．

n_{filter} : フィルタ用コーパスに属している総小説数．

$df_{filter}(w)$: フィルタ用コーパスにおいて， w が含まれている書評が存在する小説数．

4.2 フィルタ用コーパスの作成

小説を属性ごとに分類し，各属性に属する書評数が均一になるようにフィルタ用コーパスを作成する．コーパスの作成に用いる属性は，属性から予想されるキーワードが偏らないように，手作業で選択する．

小説の属性による分類は，属性キーワードを利用する．属性キーワードの定義は 2 章にて述べている．実際に用いた属性例を表 1 に示す．

表 1: 使用属性例 (15 種)

文学	学園	現代 (モダン)
超能力	ホラー	感動
SF	戦争	動物
高校生	恋愛	エンターテインメント
ハーレム	ほのぼの	ハッピーエンド

4.3 異なる小説の書評間で共通に出現するキーワードの抽出

ほぼ全ての小説の書評に共通して出現する候補語 (「設定」や「世界観」などである．以下，書評間共通のキーワード) は，(1) 式による抽出では抽出できない．これは，4.1 節で述べたように，(1) 式が特定の小説の書評のみに出現するような候補語 (登場人物名など) を抽出するための式だからである．書評間共通のキーワードの数は，それほど多くないので，事前にリスト化することが可能である．そこで，以下のようにリストを作成し，それを用いて抽出する．(2) 式による重み付けが閾値 T 以上の候補語を対象とし，その中から手作業で，書評間共通のキーワードとして相応しくないものを削除することで使用するリストを得る．(2) 式は，候補語 w が小説の本文内には出現せず，その小説の書評内のみ出現する小説の数を示す．

$$Count(w) = \sum_i^{n_{all}} ex_i(w), \quad ex_i = \begin{cases} 1, & w \in L_i \\ 0, & w \notin L_i \end{cases} \quad (2)$$

ここで， n_{all} は総小説数を示しており， L_i は小説 N_i において， N_i の書評内の候補語集合から N_i の本文内にも存在する候補語を除いた集合である．

リストを用いた抽出は，以下のように行う．リスト内の候補語中，対象小説の書評内で出現頻度が高い順に k 語をキーワードとする．このキーワード k 語中に，(1) 式による抽出キーワードと共通するキーワードが存在する場合は，それらを除去する．(2) 式により抽出した k 語のキーワードの代わりに，(1) 式による抽出キーワードの下位 k 語を除去することで，キーワード数がキーワードの上限値を上回らない様にする．

5 評価実験

小説投稿サイト (小説家になろう¹) の，2010 年 5 月から 6 月の間の感想データ 84,346 件を使用する．感想とは，データの抽出サイトにおける定義である．サイトには，感想の他にレビューが存在しており，そのどちらも本研究の定義 (3 節にて述べている) では，書評となる．レビューの数に比べ，感想の数が大きく上回っていたため，感想を用いた．

5.1 実験方法

異なる特徴を持つ 3 小説 (小説 1, 2, 3 とする) を対象とし，以下の手法で抽出を行い，比較した．特徴としては，小説 1 は有名な商業作品で用いられた台詞回しや技を真似たものが多く使われており，感想内にも，それらの作品を象徴する語 (作品名や，技名など) が頻出する．小説 2 は，感想内にキャラクターやストーリーといった語が頻出する．小説 3 は，感想内に小説中の特定のシーンを示す語が多様である．与えるキーワード数の上限を 40 とする．

手法 1 (1) 式の抽出のみを用いる．フィルタ用コーパスには，全感想を含める．

手法 2 (1) 式の抽出のみを用いる．フィルタ用コーパスは，属性数 15，感想数 300 で作成する．

手法 3 (1) 式の抽出とリストからの抽出を併用する．フィルタ用コーパスは，属性数 15，感想数 300 で作成する．

手法 3 でのリストからの抽出は， $T = 35$ ， $k = 10$ として行った．リスト作成における閾値を 35 としたところ，候補語は，274 語あり，手動で削除した結果 68 語となった．

5.2 実験結果

5.1 節に従い，実験を行う．評価としては，意味評価，意見評価，具体性評価を行った．意味評価は，キーワード中の意味が明白であるものの割合，意見評価は，キーワード中の意見でないものの割合，具体性評価は，具体性のある感想中の検索可能であるものの割合を示す．意味や意見，具体性は 3.2 節にて定義したものである．結果を表 2, 3, 4 に示す．

表 2: 意味評価 (単位: %)

	小説 1	小説 2	小説 3
手法 1	70.0	50.0	90.0
手法 2	70.0	55.0	85.0
手法 3	75.0	75.0	95.0

表 3: 意見評価 (単位:%)

	小説 1	小説 2	小説 3
手法 1	62.5	40.0	87.5
手法 2	72.5	42.5	85.0
手法 3	82.5	72.5	97.5

表 4: 具体性評価 (単位:%)

	小説 1	小説 2	小説 3
手法 1	63.8	87.2	60.7
手法 2	70.5	91.5	66.7
手法 3	77.8	95.7	82.0

5.3 κ 値

5.2 節での評価値は主観によって異なる可能性がある。3 種の小説に対して、それぞれ 3 名に評価してもらうことで 値を求める。ここで、対象とする小説と 5.2 節での対象小説は異なる。

値は、2 者間の評価の、偶然の一致を除いた一致度を表す値である。そこで、3 人の評価者から、2 者を選択する全組み合わせについて '(4) 式) を求め、そのうちの最小値を 値 ((3) 式) として用いた。

値の計算に用いた式を (3)~(7) 式に、 値の解釈を表 5 に、求めた 値を表 6 に示す。

$$= \min_{x,y} ' (x,y) \quad (3)$$

$$' (x,y) = \frac{P_o(x,y) - P_e(x,y)}{1 - P_e(x,y)} \quad (4)$$

$$P_o(x,y) = \frac{eq(x,y)}{N} \quad (5)$$

$$P_e(x,y) = \sum_{i \in V} p_i(x,y) \quad (6)$$

$$p_i(x,y) = \frac{n_i(x)}{N} \cdot \frac{n_i(y)}{N} \quad (7)$$

$eq(x,y)$: 評価者 x, y 間の評価値が等しい評価の数。

$n_i(x)$: 評価者 x の、評価値が i である数。

N : 提示キーワード数、もしくは、感想数。

V : 評価値集合。{0, 1}。

表 5: 値の解釈 [1]

Kappa	値の解釈
< 0	Poor
.00 - .20	Slight
.21 - .40	Fair
.41 - .60	Moderate
.61 - .80	Substantial
.81 - 1.0	Near perfect

表 6: 値

	意味	意見	具体性
小説 A	0.54	0.05	0.31
小説 B	0.63	0.22	0.41
小説 C	0.64	0.53	0.60

6 考察

5.2 節から、手法 1, 2, 3 の中で手法 3 により最もよいキーワード抽出が可能であることが分かった。また、4.2 節と 4.3 節の処理が狙い通りに作用していることが分かった。まず、手法 1 と手法 2 の結果を比較する。手法 1, 2 間の違いは、フィルタ用コーパスにある。これによって、影響が出てくるのは小説 1 の意見評価と具体性評価である。小説 1 は有名な商業作品を象徴する語が多用されている。例えば、「モンハン」「遊戯王」などである。これらは、二次創作として、人気のある作品であるため、手法 1 の方では、フィルタ用コーパス内で偏りが発生してしまい、抽出されにくい。手法 2 では、これらの語がキーワードとして抽出された結果、意見評価値と具体性評価値が上昇している。手法 3 の効果は明白である。小説 2 の、意味、及び、意見評価値が大幅に上昇している。また、小説 3 の具体性評価値にも上昇が見られる。これは、小説 3 にある多様なシーンに対する感想を、書評内共通キーワードなしでは、検索しきれないためである。

本研究では、抽出法に用いられていた変数に対して、より良い値を探索していない。また、表 6 から分かる、意見評価と具体性評価の主観による揺らぎに対する解決策も明らかにしていない。これらのことについては、今後の課題としたい。

7 おわりに

本研究では、レコメンデーションを目的とする書評検索システムに利用するキーワードの抽出を行った。提案手法 (手法 3) を、基準となる抽出 (3.1 節) と、それを改善することを目的とした処理 (4.2 節, 4.3 節) に分割し、処理の有無による抽出を比較実験した。実験の結果、提案手法が最も良い評価値となり、4.2 節と 4.3 節の処理の有効性を示すことができた。

参考文献

- [1] Jean Carletta, Amy Isard Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. The Reliability of a Dialogue Structure Coding Scheme. Computational Linguistics, Vol. 23, No. 1, pp. 13-31, 1997.