N-013

マンガの概要に基づく作品推薦システム
An Institutional Recommendation System based on Contents of MANGA Books

村瀬 尊好†, 柊 和佑†, 安藤 友晴†
Takayuki Murase, Wasuke Hiiragi, Tomoharu Andoh

1 背景

現在一年間で出版されるマンガの単行本は 29,394 種類あると言われており、マンガ雑誌を含めるとその数はさらに多くなる [1]。そのためマンガの利用者にとって、新旧の多量のマンガの中から自分の好みに合ったものを簡便に見つけるのは困難である。それは利用者ごとに好みの作品があり、新たにマンガを探す場合その好みをキーとして探さなければならないためである。特に好みのストーリのマンガを探す場合は、読者自身が多量のマンガを読んでみないとならないため、前もって好みに合致したものを探すのは困難である。

そこで本研究は、利用者の好みのストーリに応じたマンガを、利用者の指定した作品を基に推薦することを目的とする。これにより、読者は大量の種類のマンガの中から、ある程度容易に、好みに沿ったマンガを絞り込むことができるものと考え、システムの構築を行った。

2 既存の作品推薦システムと問題の考察

利用者の好みに応じた推薦を行うためには、利用者の好みの把握・判断、推薦対象の情報の収集・蓄積、好みを基にした検索・推薦が必要となる。本節では、既存の手法を解説すると共に、本研究が解決すべき問題について考察を行う。

・協調フィルタリングによる作品推薦システム

Amazon.co.jp[2]の商品推薦機能で用いられている協調フィルタリングは、同じ商品を購入した利用者集団を、近い好みを持つ集団と判断し、それらの利用者が高く評価している商品を推薦する手法である [3]。実際に購入されている商品を基に好みの把握を行っているため、現在人気のある商品を推薦することが可能である。

しかし、マンガの推薦では現在売れているものを提供するだけでは、充分に利用者の好みに応じた推薦ができているとは言いがたい。協調フィルタリングでは、集団の履歴により重みづけがされているため、内容は似ているが購入実績が少ないマンガについては、利用者の好みに合致していたとしても、推薦されにくいからである。

・感性キーワードによる作品検索システム

whichbook.net[4]は英国の小説作品の推薦を行うシステムである。利用者によって入力された読後の感情や作品の雰囲気を基にすることで、利用者の好みに応じた作品を検索することができる。現在は同志社大学の原田隆志らが whichbook.net の日本語版の開発を行っている [5]。この手法は、利用者が自らの好みを言語化および程度の数値化をする必要がある。しかし、マンガの場合は小説と違って感情が絵で表されているため、言語化が困難である。

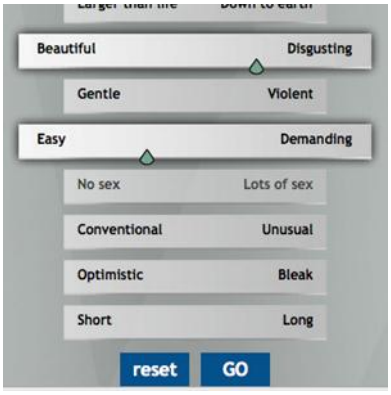

図 1　Whichbook.net の推薦サービス

†稚内北星学園大学　情報メディア学部　情報メディア学科

・タグによる検索システム

オススメマンガ参照コンテンツ [6]や YouTube[7]、ニコニコ動画 [8]といったタグによる検索システムを持つサービスも存在する。タグを使った検索システムとして林らは利用者がみたページをタグと結びつける Web ページ推薦手法を開発した [9]。これに対して佐伯らは”この手法ではユーザタグ間の関係を考慮していないため多義語と誤って重なりと認識してしまうなど、嗜好の類似性を正しく評価できない場合がある”と述べている。このため佐伯らは、ユーザの嗜好を示すプロファイルを階層化されたタグによって表現することで精度の向上を行っている [10]。佐伯らは、この手法により Web 検索の精度を向上させた。

タグによる方法を本研究で応用する場合、マンガは Web と異なり内容が機械可読文字で記述されていないため、自動でタグをつけることは困難である。このため、前述のオススメマンガ参照コンテンツは人力によってタグをつけている。利用者の価値観によってつけられたタグはストーリを考慮されているとは限らず、タグによる推薦は困難であると考えられる。

・電子番組表を基にした TV 番組推薦システム

土屋誠司らは利用者の TV 番組の視聴履歴を利用者の好みと判断し、番組を推薦するシステムの開発を行っている [11]。視聴履歴として記録されている番組の番組名、出演者、番組概要文章からキーワードを抽出し、重要な単語を抜き出して薦候補の単語の意味の一致度合から類似度を判定する。しかし、マンガの推薦では出演者などの情報が存在しないため、TV 番組のような豊富なキーワードを抽出することが困難である。なおかつ、作者や編集者の情報は、本研究が目的とするストーリには直接関係しにくい。

山崎は”記述できる文字数が制限されているため話題ごとの非常に良い要約となっている”と述べており [12]、概要文書は限られた文章量でしかない。情報量が少ないのでストーリについての記載も少ないと考えられるから本研究で使うとなると補助的な扱いとなると考えられる。番組概要文書を TV 番組のストーリとして考え、そこからストーリに基づく推薦を行うことは難しいと考えられる。また出演者の情報は作品ごとのストーリと関係しない。

以上のようにタグや感情語のような利用者ごとにばらつきのある言葉を用いると本研究が目的としているストーリに注目した推薦は困難である。また協調フィルタリングだけでは、購入者数や閲覧者数が少ない作品が推薦されにくい傾向がある。これらを複合的に使用したシステムではユーザが入力しなければならない情報が多量になる。

そこで本研究はストーリに基づくマンガの推薦を行うため、上記の問題点を解決することを目的とする。

3 マンガの概要に基づいた推薦

3-1 推薦手法の概要

本研究では、マンガの推薦をするにあたり、以下の手順を行う。

1. **推薦対象の情報の収集・蓄積**
2. **利用者の好みの把握・判断**
3. **好みを基にした検索・推薦**

まず、1. を行い、一定の記述量が存在する作品自体の解説文を収集することとした。ここでいう一定の記述量とは作品のストーリを説明するに足る記述量のことである。また、内容記述の量および閲覧者の過多によって推薦基準が変化することを防ぐため、一定以上の編集基準で作成された文章を利用することとした。ここでいう一定の編集基準とは感想主体ではなくそれを読めばマンガのストーリについて知ることができる程度の基準である。そのため、タグやジャンルといった、統制しにくい語彙群を使用することなく、文章全体を使用することとした。また検索の基となる文章の統制はとれてなくとも作品について説明できていれば問題はない。次に、2. では利用者の好みの把握を行う。3. で集めた記述を基に、利用者が好きな作品を入力すると、システムはその作品のストーリに基づき、似た傾向を持つ作品を推薦する仕組みを考案した。

本研究は、作品の内容について特定の利用者集団が作成したマンガについての記述に基づいて生成した語彙を用いることとした。そして、本システムは抽出した語彙を利用したベクトル空間モデルを用いて作品同士の類似度を判定し、推薦を行い、利用者に提示することとした。

3-2　システムの概要

システム概要は図 2 のようになっている。本システムは、まず特定の利用者集団が作成した本文データ（Contents）を収集し、類似度データ（DATA）を作成する（1）。その後、利用者より読みたいマンガの作品名のリクエストを受け（2）、類似度データを比べて、似ていると判断したマンガのリストを推薦（recommendation）する（3）。これにより、本システムはマンガの語彙を利用した推薦システムを実現している。

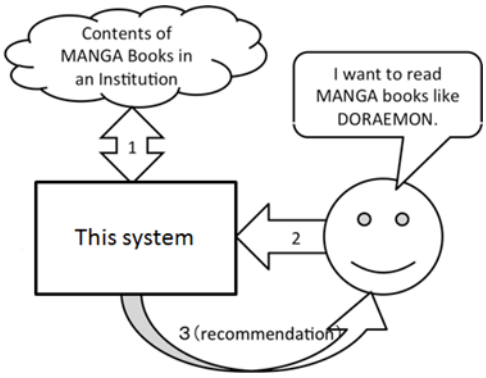

図 2　本システムの概要

4 生成された語彙を用いた推薦システム

本システムにおいて推薦には作品の内容についての本文データおよび本文データから生成された語彙が必要である。これらの語彙は以下の条件を満たす本文データから抽出する必要がある。

① 一定以上の記述量の存在

② 一定以上の編集基準の存在

条件①は作品の解説文が一定以上の記述量があることに加え、タグやジャンルとは異なり文脈が存在する本文データのことを指す。また条件②は主観的な感想ではなく事実のみを記述した本文データのことを指す。本システムはこの本文データを Wikipedia の記述のうちカテゴリに「漫画作品」を持つ本文データを対象とし、データベースに格納した。なお対象となる本文データは 10449 件である。

さらに 10449 件の本文データを形態素解析して名詞を抽出し、本文データごとに必要な語彙とした。さらに、語彙にその語彙が一つの記事中で何回使用されたか、何件の記事で使用されたかを付与し語彙リストとした。本研究では形態素解析を行うにあたって形態素解析ソフトウェア「MeCab」を使用する。Wikipedia の記事を形態素解析する前に「MeCab」がより多くの単語のパターンで形態素解析するために「MeCab」のための辞書を追加する必要がある。そこで NAIST Japanese Dictionary[13]という形態素解析用の辞書と Wikipedia とはてなキーワードのタイトルで形態素解析用の辞書を作って「MeCab」の辞書に追加している。

まず「MeCab」の辞書を作るに際して、Wikipediajawiki-latest-all-titles-in-ns0.gz[14] と http://d.hatena.ne.jp/images/keyword/keywordlist_furigana.csv[15] から Wikipedia とはてなキーワードの全タイトルを取得する。その後単語が出現する頻度を決める「コスト」を設定する。このコストは単語ごとの長さが大きいほどコストに小さな値が設定され、このコストの値が小さいほど優先して形態素解析されやすくなる。

推薦の方法は、ベクトル空間モデルを用いて作品を推薦する。まず、マンガの作品名を対象に、全語彙リストを利用した tf-idf[16]による重みづけを施した。本システムでは、これを文書ベクトルと呼ぶ。そして、利用者が指定したマンガの文書ベクトルと全ての作品の文書ベクトルのコサイン類似度を求める。この際、コサイン類似度が高い値の作品ほど、利用者が指定した作品と類似していると判断し、推薦作品として利用者に提示する。

tf-idf による重みづけを行う方法は以下の通りである。tf(term frequency)は 1 件の本文データにおけるある単語の出現頻度のことである。idf(inverse document frequency)は、単語が出現する記事数の逆数である。idf を用いることによって、Wikipedia のメタタグや「ISBN」などといった、マンガ作品の内容と関係が無く、かつ多くの記事データで使用されている単語の影響を減らし、マンガ作品の内容に関わる単語の重みづけを高める効果がある。tf-idf は tf と idf の積である。式 1 に tf-idf の式を示す。なお式中の N は記事の総数のことで、df とはある単語がどのくらいの記事で使われているかを指す。

tf-idf の例を挙げると、例えば「HELLSING」という作品で使われていた単語が「吸血鬼、ナチス、Category」と

する。この場合「Category」という単語は全記事中 10246 件の記事で使われており、このままこれを一つのベクトルとすると「HELLSING」とストーリにおける関連性のない作品との共通点となってしまうので idf の値を低く設定する。また「吸血鬼、ナチス」は作品のストーリを示す単語と考えられるので idf の値を高く設定する。また単語ごとの idf の設定は推薦する作品の検索にかかる時間を短縮するため事前に計算をしておく。

$$\mathrm{tf} \times \mathrm{idf} = \mathrm{tf} \times \left(\log_{10} \frac{\mathrm{N}}{\mathrm{df}} + 1 \right)$$

式 1：tf-idf の式

すべての作品について、全作品の単語リストに tf-idf による重みづけをしたものを要素とする文書ベクトルを作成し、指定作品の文書ベクトルとのコサイン類似度を求める。この文書ベクトルを作品の内容とする事で、文書ベクトルと文書ベクトルの角度が小さいほど類似している作品と判断した。

ベクトル空間モデルでは本文データの単語の重みを要素としたベクトルで文書を表現する。 例えば「テニスの王子様」で検索を行った場合「テニスの王子様」の本文データにある単語が「テニス．大会」だとして、単語ごとの重みがそれぞれ「45.6 , 8」とする。また「“LOVe”」という作品と「ベイビーステップ」という作品における「テニス．大会」の tf-idf の値が、「“LOVe”」ならそれぞれ「19.96 , 31.55」であり、また「ベイビーステップ」なら「39.9 , 112.158」とすると図 3 のように表現できる。コサイン類似度は同じような単語を多く含む作品に対して高い値となる。

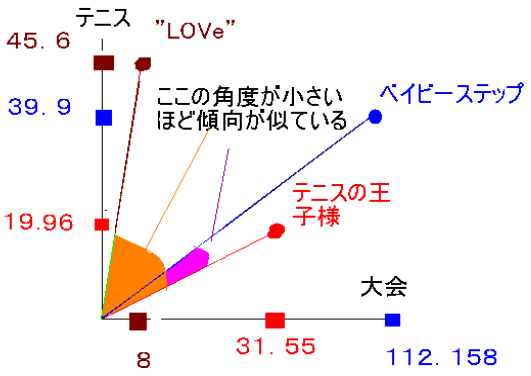

図 3　　コサイン類似度の図

作品ごとのベクトルを取得する事ができたら、この 2 作品のコサイン類似度を求め、計算された値を推薦の基準とする。また他作品に対しても同じような計算を行い、値の大きかった作品から推薦する。 なおコサイン類似度の式を式 2 に示す。

$$\cos(d_j q) = \frac{\sum_{i=1}^{m} d_{ij} \times q_i}{\sqrt{\sum_{i=1}^{m} d_{ij}^2} \times \sqrt{\sum_{i=1}^{m} q_i^2}}$$

式 2：コサイン類似度の式

5 概要から抽出した語彙による実験結果

前章で示した手法の有効性を確認するために、マンガ作品を推薦する実験を行った。実験にあたっては、利用者が好む作品を「HELLSING」、「テニスの王子様」、「ドリフターズ」の各作品とする。これらの作品と同じ傾向を持つ作品を前章の手法により取得し、これらを推薦作品とした。マンガ作品の記事データを形態素解析したところ、全作品で 282,543 種類の単語が抽出された。個々の単語の idf を調べたところ idf の値が 2 以下 4 以上のものに作品の特徴を示すような単語はそれほど多くはなく、逆にそれらをのぞかないことにより検索を行う上でのノイズとなってしまうと考えた。そこで、実験にあたっては、idf の値による単語数の絞り込みを試みた。絞り込みの基準は次のとおりである。

・idf の値による単語の絞り込みをおこなわない
・idf の値が 2 以上 4 未満の単語のみ利用する

まず値を指定しなかった場合の上位 10 件の結果を表 1、表 2 表 3 に表示する。また今回の指定作品は表 1 を「HELLSING」、表 2 を「テニスの王子様」表 3 を「ドリフターズ」とする。以下が実験結果である。

表 1：idf を絞り込まず「HELLLSING」を指定した場合

作品名	コサイン類似度
もっけ	0.3523415777
忍者部隊月光	0.3169093404
パラソルヘンベえ	0.3163268338
柔道一直線	0.307763505
ボーイズ・オン・ザ・ラン	0.3052525916
ポコニャン	0.303791957
アタック No.1	0.2958187527
BLOOD ALONE	0.290366855
愛してナイト	0.290366855
ワンサくん	0.2902300676

表 2 idf を絞り込まず「テニスの王子様」を指定した場合

作品名	コサイン類似度
学校のおじかん	0.78915740855209
不思議の国の千一夜 ？ ヘンデク★アトラタン物語?	0.7688979499660
ぷちモン-Petit Monster-	0.76829138302682
クローバー	0.7466918114780
日露戦争物語	0.73970772729341
交通事故鑑定人 環倫一郎	0.73476875489185
青春ビンタ!	0.73017228020481
ちゃお	0.72552823691391
射雕英雄伝	0.72542309477502
天才柳沢教授の生活	0.71725901488408

表 3：idf を絞り込まず「ドリフターズ」を指定した場合

タイトル	コサイン類似度
ツギハギ漂流作家	0.232588737186882
パンプキン・シザーズ	0.17588727347202
BASTARD!! 暗黒の破壊神	0.17436723838155
漂流教室	0.1571357595120 90
きゅーきゅーキュート!	0.156663896278828
妖怪のお医者さん	0.154460584788590
対応音声出力	0.153831132481614
スレイヤーズ	0.153352116563233
殿といっしょ	0.152581135565725
キングダム	0.151893687858358

次に idf の値が 2 以上 4 以下の単語を取得して作品ごとのコサイン類似度を計算した。指定作品は、表 4 は「HELLSING」、表 5 は「テニスの王子様」、表 6 は「ドリフターズ」とする。なお表 5 における（リダイレクト記事）および（カテゴリを格納した記事）とは、表記ゆれ、およびシリーズ名変更を吸収するために設置された Wikipedia 内の記事の事であり、単語数が少なく、作品名だけが書かれていることが多いため、コサイン類似度が高くなってしまう傾向にある。

表 4：idf を絞り込んで「HELLSING」を指定した場合

タイトル	コサイン類似度
パラソルヘンベえ	0.50612954792931
忍者部隊月光	0.4569556440898
もっけ	0.44323400970152
柔道一直線	0.43994899900450
ポコニャン	0.429134905025422
愛してナイト	0.42748944473576
ボーイズ・オン・ザ・ラン	0.3873791584386
アタック No.1	0.38223181997936
ワンサくん	0.36876806815145
プラトニックチェーン	0.363164261883249

表 5：idf を絞り込んで「テニスの王子様」を指定した場合

タイトル	コサイン類似度
放課後の王子様	0.63001719409582
（リダイレクト記事）	0.34668252344999
（カテゴリを格納した記事）	0.34288950195921
COOL-RENTAL BODY GUARD-	0.23021466014011
ベイビーステップ	0.12285252504269
燃える V	0.09964184170451
しゃにむに GO	0.09565262386827
太臓もて王サーガ	0.09512797724652
STAY GOLD	0.08675415864787
"LOVe"	0.0840045711337

表 6：idf を絞り込んで「ドリフターズ」を指定した場合

タイトル	コサイン類似度
ツギハギ漂流作家	0.36877752004367864
内閣総理大臣 織田信長	0.17866331083655812
MISTER ジパング	0.17035006648357387
秀吉でござる!!	0.15619408378969432
夢幻の如く	0.1539561093031851
センゴク外伝 桶狭間戦記	0.152138671609567
炎のニンジャマン	0.15051909137157798
センゴク	0.14074667974525587
殿といっしょ	0.13504013179247787
戦國ストレイズ	0.1285996914856118

6 結論

コサイン類似度による推薦を行う際、システムが取得する単語の idf の値の範囲を指定しなかった場合、表 1 の指定作品である「HELLSING」という作品で推薦された作品を筆者が「HELLSING」の内容を踏まえて上位 10 件に入った作品の Wikipedia の記事を調べた場合近い作品が推薦される一方、「HELLSING」の内容と一致しない作品も見られた。また「テニスの王子様」が指定作品である表 2 と「ドリフターズ」が指定作品である表 3 も同様の結果であった。以上の結果から idf をしぼりこまない場合適した結果を出すことは難しいと考える。これは、Wikipedia 特有の記号や、出版社の URL などといった文字列が特徴としてあらわれてしまうためであると考えられる。

取得する単語の idf の値が 2 以上 4 以下の場合である表 4、表 5、表 6 の結果をそれぞれ指定作品の内容に基づいて筆者が調べたところ、「テニスの王子様」ではテニス系統のマンガが多く推薦され、「ドリフターズ」では戦国時代をテーマとしたマンガが多く推薦されたが、「HELLSING」の場合は絞り込みを行わなかった場合と大差はなかった。

7 まとめと今後

本研究では利用者が指定した作品と類似した作品を推薦することを目的とし、システムの構築および実験を行った。本システムの特徴は、本文概要を利用することで従来のタグやジャンルといった、統制が比較的困難な語彙群と併用が可能な語彙リストを利用したことである。しかし、概要だけを利用した実験では、結果として成功する場合と失敗する場合があった。原因としては、tf-idf で用いた重みづけの検討が不十分であった事が考えられる。しかし、本研究で用いた語彙は 28 万語以上あり、システム自体も検証を進めながらの実装であったため、重みづけを変更するたびに約半日以上の再計算が必要であった。今後はアルゴリズムおよび計算機環境の改善をすすめ tf-idf の重みづけの検討を進めていきたいと考えている。

また、Wikipedia 特有の記号や記事内の URL といったノイズの存在も考えられる。tf-idf を用いることでノイズの対策をとれると思っていたが結果的に回避できなかったため今後は、どのような言葉がノイズとなるのか、検討を行い事前に排除するつもりである。さらに、本研究では Wikipedia の本文データをそのまま利用していたため、今後は表記ゆれや不必要な記述の削除、構造を考慮した語彙の抽出なども検討し、導入した上でどの程度効率が向上するか検討していきたいと考えている。

そして、今後は利用者によって付与されたタグやジャン

ルといった、従来から存在する手法と合わせて推薦を行いたいと考えている。その場合、どのように検証を行うかの方法も考察する必要があるだろう。

なお、これらの改善で計算速度が向上された場合、Web などを利用して利用者からのリクエストを受けるサービスへ発展させたいと考えている。近年、電子書籍によるマンガの閲覧も一般的になりつつあり、オンラインを利用したシームレスな検索と購入のための手法が求められている。本研究は、そのような用途にも利用できるのではないだろうか。

参考文献

1)電通総研 . 情報メディア白書 2012 . 2012 年 1 月 13 日第 1 版発行, pp60-61.

2)Amazon.com ."Amazon.co.jp" .Amazon.co.jp . 2012 年 4 月 18 日 http://www.amazon.co.jp/ ,(2012 年 2 月 16 日確認)

3)Greg Linden, Brent Smith, and Jeremy York . Amazon.com Recommendations :Item-to-Item Collaborative Filtering, IEEE Internet Computing, Jan. /Feb.2003

4)原田隆史 .感性パラメータを用いた類似する小説の提示. 情報知識学会誌 .vol.21,No.2, pp.291-296,2011.

5)Whichbook.net ."Whichbook | A new way of choosing what to read next" .Whichbook.net .2012 年 4 月 16 日 http://www.whichbook.net/ ,(2012 年 2 月 17 日確認)

6)株式会社オーバーライド ."人気漫画から隠れた名作までオススメマンガがすぐ見つかる!". オススメマンガ参照コンテンツ . 2012 年 4 月 16 日 .http://www.osman.jp/. (2012 年 2 月 15 日確認)

7)Youtube, LLC ."Youtube – Broadcast Yourself" . Youtube .2012 年 4 月 18 日. http://www.youtube.com/ ,2012 年 4 月 18 日確認

8)株式会社 ニワンゴ. "ニコニコ動画(原宿)". 2012 年 4 月 18 日. http://www.nicovideo.jp/ ,(2012 年 4 月 18 日確認)

9)Shuhei Hayashi, Yuuki Inoshita,and Satoshi Fujita. An efficient web page recommendation based on preference footprint to browsed page. 第 5 回情報科学ワークショップ,2009.

10)佐伯祐太,林周平,井下雄樹,藤田聡 . 知識概念に着目したユーザの分類に基づくパーソナライズド Web 検索システムの提案 .情報処理学会研究報告 .2009-DPS-141 .30 ,1-7

11)土屋誠司,佐竹純二,近間正樹,上田博唯,大倉計美,蚊野造,安田昌司 .TV 番組推薦システムの構築とその有用性の検証 . 情報処理学会研究報告 .2006(3), 95-102, 2006-01-13

12)山崎智弘 .強連結成分分解を利用した電子番組表からの話題抽出 .DBSJ Journal, Vol.7, No.1, pp.1-6

13)Hideki Yamane ほか. "NAIST Japanese Dictionary プロジェクト日本語トップページ". NAIST Japanese Dictionary .2012 年 3 月 20 日 http://sourceforge.jp/projects/naist-jdic/ ,(2012 年 2 月 15 日確認)

14)Index of /jawiki/latest/ .2012 年 6 月 16 日. http://dumps.wikimedia.org/jawiki/latest/, (2012 年 6 月 27 日確認)

15)はてなダイアリーキーワードふりがなリストを公開しました-はてなダイアリー日記 .2006 年 9 月 22 日 http://d.hatena.ne.jp/hatenadiary/20060922/1158908401 (2012 年 6 月 27 日確認)

16)Luhn, Hans Peter. 1957. A statistical approach tomechanized encoding and searching of literary information. IBM Journal of Research and Development 1(4):309–317. 133,527