ニコニコ動画 API を用いた特徴語分析による動画推薦システム

2012SE167 守川峻耶　2012SE209 尾崎俊介
2012SE212 酒井良輔　2012SE266 牛田泰樹
指導教員：河野浩之

1　はじめに

近年，インターネットの普及に続き，スマートフォンの普及等によりソーシャルメディアの利用率が全体で 6 割越えと拡大している [1]．ソーシャルメディアの一つである動画閲覧サイトの代表として，ニコニコ動画，YouTube，Dailymotion 等があるが，今回我々はコメント分析に重点を置くため，時間軸毎にコメントが投稿されるニコニコ動画を分析することにした．ニコニコ動画では，投稿動画数が 1200 万以上と莫大な量となっているため視聴したい動画を発見することは極めて困難である．ニコニコ動画ではランキング上位の動画のみが多く見られる傾向にあるので，中小規模のユーザが楽しんでいる動画は発見されにくいという問題がある．

本研究では，ランキングに掲載されていないため多数の人には知られていないが，視聴した際に多くの人の興味・関心が湧く動画コンテンツの発見手法を提案することを目的とする．また，YouTube では動画を端的に表した「タグ」を投稿者が付与することに対し，ニコニコ動画ではタグを視聴者が付与する．よって，ニコニコ動画のタグに着目することは視聴者の感覚に近い動画の発見に繋がると考える．今回コメント，タグの双方を分析することにより，よりユーザの嗜好にあった動画を推薦する．そこで我々は動画発見の方法として，平澤らの「もっと評価されるべき」タグに着目した手法を用いることにする [2].「もっと評価されるべき」タグが付与された動画にコメント分析 [3] を行い，特徴語を抽出し，類似度を付与することによって，ユーザの嗜好に合った動画推薦を目指す．特徴語抽出には「ばっちりサーチ.net」[4] を用いる．

本研究の構成を以下に示す．2 章では，従来の動画推薦システムおよびニコニコ動画に関する先行研究について述べる．3 章では，先行研究に対する問題点の解決方法，提案する推薦システムの構成を述べる．4 章では，様々な API を用いてメタデータの取得を行う．取得したメタデータから特徴分析を行い，類似度計算の結果からユーザに動画を推薦する．5 章では，本推薦システムについて実験・評価をする．6 章では，本研究のむすびについて述べる．

2　動画推薦に関する先行研究

2.1　ニコニコ動画のログデータを用いたソーシャルノベルティのある動画の発見に関する研究 [2]

平澤らは，「社会的には知られていないが，より多くの人が興味・感心を持つコンテンツ」をソーシャルノベルティのある動画と定義し，コメントやタグ，再生数等に注目することでソーシャルノベリティのある動画を特徴づける特性として，以下の 3 つの特性に注目した [2].

1. コメント特性：コメントに注目
2. 動画内容特性：タグに注目
3. 視聴行動特性：再生数，コメント数，マイリスト数注目

これらの特性に基づき，「もっと評価されるべき」タグがソーシャルノベルティのある動画の正解データとして利用できることを確認した．また，「もっと評価されるべき」タグが付与されている動画のコメントの特徴語を抽出し，その特徴語はポジティブな言葉が多いことを確認した．一方，「もっと評価されるべき」タグの有用性は確認できたが，システムとしては未実装である．

2.2　視聴者のコメントに基づく動画検索および推薦システムの提案 [3]

中村らは，視聴者の反応に基づく動画の検索と推薦を可能とするシステムを提案した [3]．ある人物が活躍しているシーンを「活躍シーン」として定義し，各登場人物の活躍シーンと活躍の大きさを，動画の再生時刻に沿って付与されたコメントを分析することで求めた．さらに，「笑える」や「泣ける」など視聴者の動画に対する印象情報に基づいた動画検索システムを実装した．

活躍パターンや印象情報を指定した検索には，一定の需要があることが確認できた．また，視聴者の反応に基づく動画の推薦では，ストーリー性のある動画ほど精度の高い推薦が行えるという知見が得られた．一方， 検索や推薦において，精度の検証が不十分であるために精度の検証，改善を行う必要がある．

3　動画推薦システムの提案

3.1　先行研究の問題点に対する改善提案

本節では，先行研究の問題点に着目し，その解決手法を提案する．まず，以下のような問題点を挙げた．

- 「もっと評価されるべき」タグがソーシャルノベルティのある動画発見に有用であることを示したが，システム化に至っていない
- 登場人物が定まっていない動画に関しては精度が不十分

以上の問題点に対する次の二点の改善案を提案する．

- 「もっと評価されるべき」タグを用いたシステムの実装，コメントにおける特徴分析により，精度の向上を図る
- 登場人物に左右されない推薦にするため，ニコニコ動画における「タグ」を用いた手法を採用し，分析する特徴語の種類を大幅に増加させる

3.2　動画推薦システムの構成図

本節では，上記の問題点と提案を考慮し，新たな動画推薦システムを構築する．

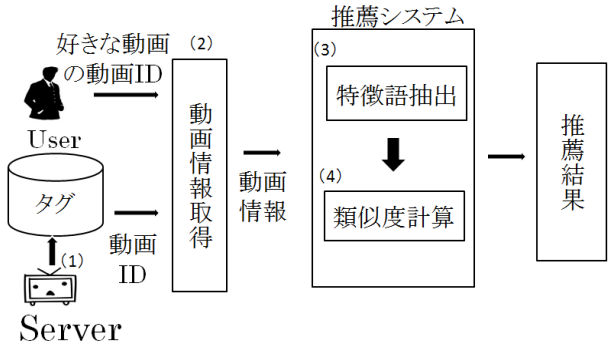

図 1　動画推薦システムのアーキテクチャ

図 1 は本研究で提案するシステムのアーキテクチャ図である．図 1(1) でニコニコ動画のサーバから「もっと評価されるべき」タグの付与された動画のみを抽出する．図 1(1) で取得した「もっと評価されるべき」タグの付与された動画の動画 ID を元に図 1(2) で動画情報を取得する．動画情報とは，再生数，マイリスト数，スレッド ID，コメントといったメタデータのことを指す．図 1(3) では，図 1(2) で取得した動画のコメントをコメント分析し，特徴語を抽出する．図 1(4) では，データベース内にある動画コメントの特徴語とユーザのお気に入りの動画との特徴語との類似度を計算し，推薦結果を出力する．

3.3　ニコニコ動画 API・ツール

本研究の推薦システムに用いるコメント分析のために必要な値が，図 1(1) の動画 ID，図 1(2) のスレッド ID，動画へのコメントである．図 1(1) においてニコニコ動画における動画 ID を取得するために RSS を使用した．図 1(2) においてニコニコ動画におけるスレッド ID を取得するために getflv，動画 ID を取得するために msg を使用した．また，図 1(2) において動画タイトル，再生数，マイリスト数を取得するためには，getthumbinfo を使用した．getflv，msg，getthumbinfo は全てニコニコ動画 API である．

ばっちりサーチ.net は，グルメやイベントなどの様々なジャンルを検索できる web サービスを運営している web サイトである．サービスの一つに「ニコニコ動画コメント@ばっちりサーチ.net」[4] がある．これは，ニコニコ動画に投稿されている動画のコメントを分析し，コメントの傾向や動画の評判などを出力することのできる WEB サービスである．この WEB サービスでは，コメント分析モジュールのソースコードを配布しており，初めからニコニコ動画特有の未知語に対応している点で MeCab を用いるより，開発コストが低く，よりニコニコ動画の分析に適していると判断し，このモジュールを用いる．

4　動画推薦システムの実装

4.1　動画 API を用いたシステムの流れ

本動画推薦システムの流れを図 2 に示す．本研究での実装環境は Windows7 32bit，Intel Core i5-2520M，メモリ 4GB 及び Linux Mint17.2 64bit，AMD FX-8350 Eight-Core，メモリ 16GB である．使用言語は python，sqlite3，javascript である．主要なプログラムは約 500 行からなる．

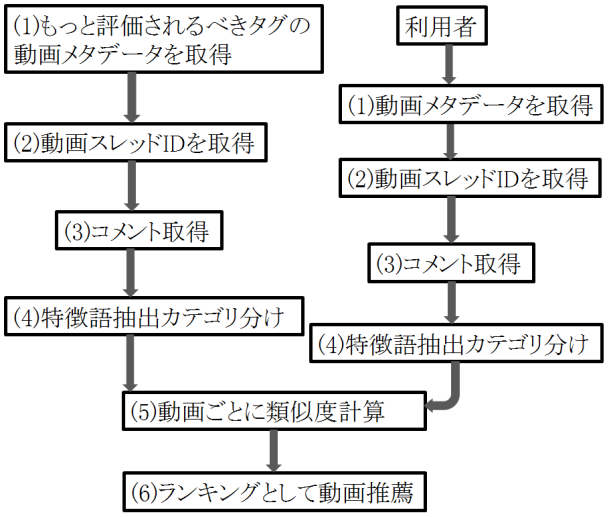

図 2　動画推薦システムの流れ

図 2(1) では，「もっと評価されるべき」タグの付与された動画の動画 ID を RSS から取得し，getthumbinfo を用いてメタデータを取得する．図 2(2) では，動画情報を getflv を用いて図 2(1) で取得した動画のスレッド ID を取得する．getflv は，RSS により取得した動画 ID を用いてコメントを取得するために必要なスレッド ID を取得する API である．図 2(3) では，msg を用いて動画内のコメントを取得する．msg は，getflv により取得した動画情報のスレッド ID を用いて動画のコメント全てを取得する API である．図 2(4) では，msg によって取得したコメントを「ばっちりサーチ.net」のモジュールを用いてコメントの特徴分析を行う．これは，入力したコメントをモジュール独自の辞書から特徴分析を行うため，ニコニコ動画特有の未知語にも対応可能である．図 2(5) では，msg により取得したコメントのコサイン類似度を用いてユーザーの好きな動画とデータベースの動画との類似度を計算する．図 2(6) では，図 2(5) で求めた数値をランキング化して上位 3 位まで表示する．

4.2　ニコニコ動画データベース構築

本動画推薦システムでは，ユーザにお気に入りの動画を入力してもらい，その動画のコメント内の特徴語から我々が構築したデータベース内の動画と類似度を求め推薦を行う．データベース構築には，sqlite3 を用いて構築しているが，コメント分析には，javascript を用いている．なお，

データベースは図 1(1) で取得するメタデータ，図 2(3) で取得するコメント特徴数をそれぞれ格納するための 2 つのデータベースを作成する．

```
create table movie (
  ID varcher(8),
  title varcher(1000),
  views integer,
  mylist integer,
);
```

図 3 データベーステーブル

図 3 は図 1 の (1) で取得するメタデータを格納するデータベースである．ID が RSS よって取得した動画 ID，title，views，mylist がそれぞれ getthumbinfo によって取得した動画タイトル，再生数，マイリスト数である．同様に，図 2(3) で取得するコメント特徴数を格納するデータベースも作成する．属性は movie テーブルと同様の動画 ID に加え，動画毎のコメント特徴数を 21 種類作成する．しかし，テーブルの中の「url」,「unknown」は類似度計算に悪影響を及ぼすため，今回はこの 2 つを省いた 19 個とする．データベースは，dictionary 形式でコメント特徴数を格納し，各属性に挿入する．

4.3 API を用いたメタデータの取得

今回我々は，最新バージョンである「RSS2.0」を用いる．また，RSS と同様に更新情報を知らせる API である Atom(Atom Syndication Format) でも代用可能である．結果は，xml 形式で出力される．図 4 に「もっと評価されるべき」タグが付与された動画を入手するプログラムを示す．図 4 のプログラムは RSS を取得し，動画 ID を配列 ids に格納するプログラムである．uri に記載されている SEARCH_TYPE，keyword，SORT_TYPE，npage はそれぞれ，タグ，もっと評価されるべき，f，ページ数を示す．SORT_TYPE=f は最新の投稿日時を示す．また，最後の行は，findall で取得したい link の部分を指定し，さらに，text として読み込み，/で区切られている一番右の部分を取得する．

```
while((nrow_bef!=len(ids))and
                    (npage<MAX_PAGENUMBER)):
 npage+=1
 uri = 'http://www.nicovideo.jp/%s/%s?sort=%s
 &rss=2.0&page=%d'%(SEARCH_TYPE,keyword,
                              SORT_TYPE,npage)
 ids+=map((lambda x:x.text.rsplit('/',1)[1]),
          rss.findall('./channel/item/link'))
```

図 4 動画 ID 取得するプログラム

今回我々が取得する値は動画 ID，動画タイトル，再生数，マイリスト数である．RSS で取得する動画 ID は，既に消去された動画も含まれているので getthimbinfo で再び動画 ID を取得し，現存する動画のみに絞る．図 5 のプログラムは図 4 で動画 ID を格納した ids を利用し，動画 ID，タイトル，再生数，マイリスト数を入手するプログラムの一部である．最後の行の meta0 にはそれぞれタイトルを格納している．同様に再生数は meta1，マイリスト数は meta2，動画 ID は ids2 に格納する．

```
for x in ids:
 uri2='http://www.nicovideo.jp/api/
                          getthumbinfo/' + x
 meta0[len(ids):]+=map((lambda x:x.text),
           rss2.findall('./thumb/title'))
```

図 5 動画メタデータの取得プログラム

getflv は，他のニコニコ動画 API と違い，ニコニコ動画にログインする必要がある．そのため本研究用のアカウントを作成した．getflv の結果は，WWWForm 形式で返却される．図 6 にスレッド ID，メッセージサーバ URL を取得するプログラムである．このプログラムは取得した動画 ID を y に格納し，for 文を使用し，正規化を用いて 4 行目の p にスレッド ID，6 行目の l にメッセージサーバ URL を格納するプログラムである．

```
for y in ids2:
 uri3='http://flapi.nicovideo.jp
                            /api/getflv/' +y
 p+=re.compile('thread_id=(\d+)').
                             findall(string)
 l+=re.compile('ms=(http.*%2F.*%2F.*%2F).
                   ms_sub').findall(string)
```

図 6 スレッド ID メッセージサーバ取得プログラム

図 7 に msg を用いてコメントを入手するプログラムを示す．図 7 のプログラムは 1 行目に zip を用いてメッセージサーバ URL の配列 l，スレッド ID の配列 p に対する for 文であり，2 行目で %2F を「/」，%2F を「:」に置換する．また，URL を開いて，xml 文書の chat 部分にコメントが記載されているため，その場所を取得し，最後の行の meta3 に格納するプログラムである．

```
for v, w in zip(l, p):
 v2=v.replace("%2F","/").replace("%3A",":")
 uri4='%sthread?version=20090904&thread=%s
                       &res_from=-100'%(v2,w)
 meta3=map((lambda x:x.text),rss4.
                           findall('./chat'))
```

図 7 コメント取得プログラム

4.4 コサイン類似度による推薦動画の決定

次に，図 2(4) でそれぞれの動画に投稿されたコメントから「ばっちりサーチ.net」のモジュールによって特徴語分析された語にコサイン類似度を用いることにより，動画毎の類似度を求める．コサイン類似度とは，文書間の類似度を計算する手法の一つである．

$$C(x) = \{c_{1_x}, c_{2_x}, c_{3_x}, ..., c_{n_x}\} \tag{1}$$

式 (1) は動画 x の各分類のコメントの出現頻度を要素 c_{n_x} として持つ特徴ベクトル $C(x)$ である．また，n はコメントのカテゴリ数であり，本研究で分類されるカテゴリ数は $n = 19$ とする．次に，動画同士の類似度計算を行う．コサイン類似度の式 (2) を以下に示す．

$$S\{C(x), C(a)\} = \sum_{i=1}^{n} c_{i_x} c_{i_a} \tag{2}$$

$0 \leq S\{C(x), C(a)\} \leq 1$ であり $S\{C(x), C(a)\}$ が 0 に近いほど動画 x と動画 a は類似せず，1 に近いほど類似している．これをユーザーが指定した動画 x とデータベース内の動画 a の特徴ベクトルに適用し類似度になる．

5 評価実験

本章では，本動画推薦システムの性能評価を示す．データベース構築時間は性能評価の大きな指標である．よって，本推薦システムの性能評価のために，データベース構築の際の動画の取得可能な限界数，動画取得数による費やす時間を評価する．本研究では，再生数，マイリスト数，動画タイトル，スレッド ID を含めた動画情報を 300 件を取得し，1 動画につきコメント 100 件を取得した．動画 1 件の取得にかかる時間は 37 秒，今回取得した 300 件を取得するのに費やした時間は 11,100 秒であった． 動画 1 件あたりに 37 秒の時間を用いたのは，ニコニコ動画 API のメタデータ取得に時間を費やさなければメタデータを完全に取得することができないので，プログラムに time.sleep コマンドを加えたためである．

本動画推薦システムのデータベース量を増やすことにより類似度順位が上下した．データベースは多ければ多いほどユーザのお気に入り動画との比較対象が増え，ユーザに興味を持たれるような動画の発見の可能性が高まる．本動画推薦システムに対し，「オセロをボッチで遊ぶ方法作ってみた！」というキーワードを入力した結果，【将棋サッカー】エキシビジョン 山口恵梨子女流球王 vs 中田善晴九段，ブロックマンエグゼ 26 2 第 1 話，【二人で実況】2015 年も彼女できませんでした part3【アマガミ】が推薦され，順に類似度は 0.4866，0.4850，0.4841 であった．ユーザが入力した動画の情報を取得するプログラムの実行時間とコサイン類似度プログラムの実行時間を計測した．表 1 より，100 件，200 件，300 件と動画数が増えても計算時間の増加はほぼ一定であった．「もっと評価されるべき」タグが付与された動画は 289,026 件 (2016/1/14 閲覧) あるので，件数増加の基づく計算時間の増加を予測し，表 1 に記す．このタグが付与された全動画を計算するのに約 6.6 秒かかることが予測される．また，動画情報を取得するプログラムの実行時間の平均は約 3.03 秒であり，全体の実行時間は約 9.63 秒となるため有用な動画推薦プログラムである．

表 1 類似度計算時間

	動画数 (件)	時間 (秒)
計算時間	100	0.0025
	200	0.0046
	300	0.0068
予測時間	1,000	0.023
	10,000	0.23
	100,000	2.3
	200,000	4.6
	289,026	6.6

6 むすび

本研究では，マイナーであるがユーザの興味を惹くような動画の発見を目的とし，ニコニコ動画における「もっと評価されるべき」タグが付与された動画におけるコメントの特徴語による推薦システムを実装した．データベース内には，ニコニコ動画 API を用いて 300 件の動画メタデータと動画 1 件につき動画内のコメントを 100 件を格納した．また，データベース内の動画コメントを特徴語分析し，特徴語テーブルに格納した．そして，格納した特徴語を類似度計算した．類似度計算にかかる時間は比例的に増え，「もっと評価されるべき」タグが付与された全動画に対する計算時間 9,63 秒より有用な動画推薦プログラムが構築できた．

参考文献

[1] 総務省:「平成 26 年情報通信メディアの利用時間と情報行動に関する調査報告書」の公表 〈http://www.soumu.go.jp/menunews/s-news/01iicp0102000028.html〉(2016 年 1 月 11 日 閲覧).

[2] 平澤真大, 小川祐樹, 諏訪博彦, 太田敏澄, “ ニコニコ動画のログデータに基づくソーシャルノベルティのある動画の発見手法の提案, ”情報処理学会論文誌, Vol.54, No.1，pp.214-222，2013.

[3] 中村聡史，山本岳洋，田中克己, “ 視聴者のコメントに基づく動画検索および推薦システムの提案, ” 情報処理学会論文誌，Vol.52，No.12，pp.3471-3482，2011.

[4] ニコニコ動画コメント@ばっちりサーチ.net 〈http://nicomment.batch-re-search.net/〉(2016 年 1 月 12 日 閲覧).