

# ユーザーイメージ収集インターフェイスの開発

戸根木 千洋<sup>†</sup> 湯浦 克彦<sup>†</sup>

静岡大学大学院 情報学研究科<sup>†</sup>

## 1 はじめに

近年、CGM (Consumer Generated Media) と呼ばれる消費者形成型メディアの影響力が増大したことにより、収集可能なユーザのロコミの量が急激に増加した。また、CGM を用いることで、イメージ調査をリアルタイムに自動的に行うことが期待できる。しかしながら、イメージ調査を行うためには定型データのイメージの方が分析しやすいが、CGM で扱われるイメージの多くは文章などの非定型データである。本稿では、定型データのイメージ収集を促進するための手法の1つとして、対話型イメージ収集インターフェイスを提案する。

また、提案インターフェイスを用いたシステムを開発することで実現可能性を検証している。イメージを収集するだけでなく、収集したイメージを自動分析し、その結果をユーザ向けのコンテンツとして表示する一連のプロセスを試作したので報告する。

## 2 対話型イメージ収集インターフェイス

### 2.1 インターフェイスのコンセプト

本稿で提案するイメージ収集インターフェイスでは、最終目標としてユーザーイメージがよりマーケティングに活用されるようになり、商品提供側とユーザの意思疎通を促進することを掲げている。この目標実現までの過程において、商品提供側とユーザ側の双方の負担を減らすことを課題とし、解決策を以下に示す。

- 定型のユーザーイメージを収集する
- ユーザが楽しく入力できる
- 収集から分析・活用まで一貫して提供する

商品提供側には、分析しやすい定型イメージを収集し、部分的に自動分析する機能を提供する。ユーザ側には、楽しく利用できる CGM としてイメージを入力する機能と、分析結果を表示するコンテンツを提供する。イメージ図を図 1 に示す。

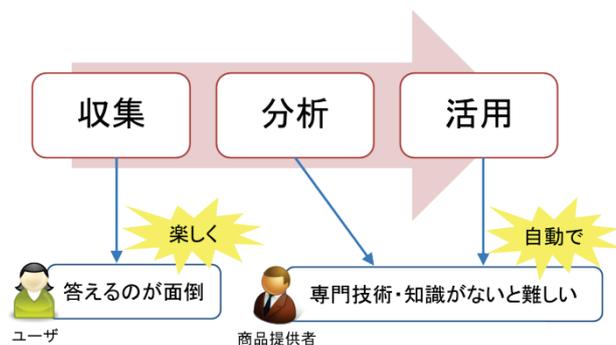


図 1 イメージ収集の課題と解決策

### 2.2 参考にした既存システム

ユーザとの対話によって、情報を収集・蓄積していくシステムで注目されたものの一つとしてアキネーター[1]が挙げられる。アキネーターは、ユーザが思い浮かべた人物をランプの魔人風のキャラクターが当てていくゲームサイトである。ユーザが利用するたびに、データベースにデータが蓄積され、正解の精度が上がっていく。開発者の手を離れても、ユーザによってシステムが学習していく仕組みになっている。このように、ゲームを提供しながら、データベースにデータを蓄積し続けることができるビジネスモデルに着目し、ユーザーイメージ収集においても同様のビジネスモデルを活用することができると考えた。

### 2.3 イメージ収集インターフェイスの流れ

まず、イメージ収集インターフェイスの対話部分の流れを図 2 に示す。

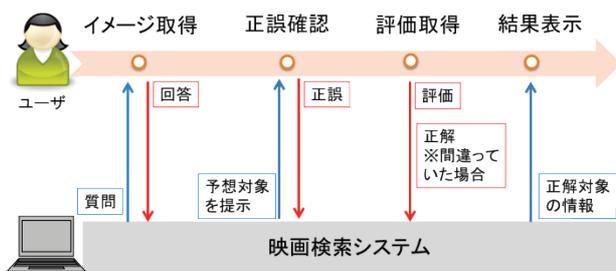


図 2 : ユーザとシステム間の対話の流れ

対話の流れは4つのフェーズで構成される。

- ① システムの質問に対してユーザが回答することでイメージを取得する（複数回）
- ② システムが取得したイメージを分析することによって対象を絞り込み、その予想が正しいかどうかを答えてもらう
- ③ 対象へのユーザ評価を取得する
- ④ ユーザが思い浮かべていた対象についての情報を表示する

フェーズ②における分析手法については 2.4 で詳細を説明する。

## 2.4 分析アルゴリズム

ユーザが思い浮かべた対象を絞り込む手法として、ベイズ推定を用いている。ベイズ推定とは、ある証拠に基づいて、その原因となった事象を推定するための確率論的方法である。フェーズ①におけるイメージ取得ごとに、各対象の該当度合をベイズ推定によって計算する。

また、イメージは固有のものではなく、人により異なるものであるという問題点には、集合知を用いることで対応する。集合知とは、多くの人による大量の情報の寄せ集めの集計のことである。[2]

## 3 提案インターフェイスのシステム化

### 3.1 映画イメージ収集システムの機能

提案インターフェイスの実用性を検討するために、映画イメージ収集システムを試作した。図3にシステム利用画面の一部を示す。

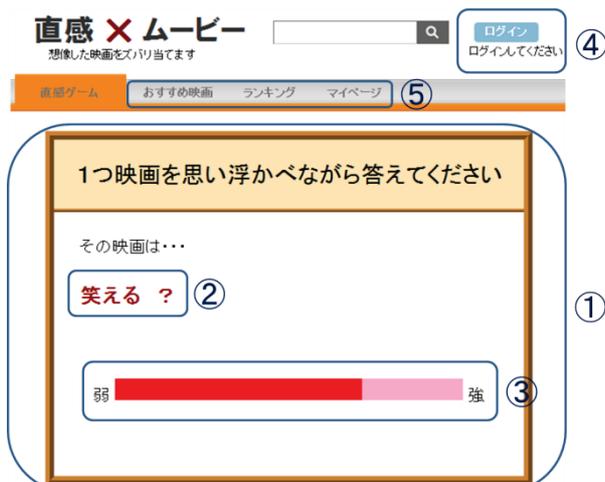


図3 映画イメージ収集システム画面

- ① 提案インターフェイスを用いた対話（2.3のフェーズ①の部分）
- ② システムが提示する質問

- ③ ユーザの回答（当てはまっている度合）
- ④ ログインすることでユーザごとの情報取得
- ⑤ ユーザに分析結果や収集した情報を公開するためのコンテンツとして「おすすめ映画、ランキング、マイページ」を提供

### 3.2 開発環境

システム開発には、Google App Engine[3]というアプリケーション実行サービスを用いている。また、開発ツールにはEclipse[4]を使いJavaによって記述した。

初期データとしてYahoo!映画[5]のイメージワードランキングの情報を参照した。

### 3.3 考察

システム化の実現によって、今回の開発環境ではデータ量に限界があることに気付いた。データ数に比例した分析量が必要になるため、データ数が多いと応答に時間がかかってしまう。

Yahoo!映画では4万件以上の作品を掲載しているが、500件以上の作品を格納すると、質問の応答時間が5秒以上になり支障をきたす。速度の高速化を図る、もしくは、小規模から中規模のデータ量のシステムでの利用に限定するといった施策が必要であることがわかった。

## 4 まとめと今後の課題

本稿では、イメージを収集するインターフェイスを提案し、収集・分析・活用を一貫して行う映画イメージ収集システムを試作した。

今後の課題としては、システムの使用評価を行い、それにより明らかになった問題点への解決策を検証する。

また、映画以外の分野においてのシステム化の検証をするとともに、システム利用者が最小限の入力や設定を行うだけで利用目的に合ったシステムが使用できるように、変更が容易なインターフェイスへの改善が必要である。

### 参考文献

- [1] アキネーター <http://jp.akinator.com/>
- [2] ジェームズ・スロウィッキー「『みんなの意見』は案外正しい」（2006/1/30）
- [3] Google App Engine <http://code.google.com/intl/ja/appengine/>
- [4] Eclipse <http://www.eclipse.org/>
- [5] Yahoo!映画 <http://movies.yahoo.co.jp/>