

## ユーザーから評価されるスマホアプリレビューの予測手法

## A prediction method of highly rated reviews in smartphone applications

古波津 巧真<sup>†</sup>  
Takuma Kohatsu太田 修平<sup>‡</sup>  
Shuhei Ota石井 信明<sup>‡</sup>  
Nobuaki Ishii

## 1. はじめに

現代のスマホアプリのプラットフォームでは、ユーザーがスマホアプリに対して、自由に評価や意見を書き込めるレビュー機能がある。例として Google Play[1]では、ユーザーは星 1~5 の評価と、そのスマホアプリに対して自由記述で評価をつけることができる。これによりユーザーは、スマホアプリの総合評価および全レビューを参照することができる。

Apptentive [2]の調査では、ユーザーがスマホアプリをインストールする際、79%のユーザーがダウンロード前にレビュー文章を参考にしている。ユーザーは、レビューに対して「参考になった」とレビュー自体を評価できる。この「参考になった」と評価された回数（以下、参考数）は一般に公開されており、参考数の多いレビューは他のユーザーに有益な情報を含んでいると言える。

古くから投稿されているレビューは、ユーザーに見られる機会が多く、参考数が多くなりやすい。しかし、比較的新しいレビューおよび参考数が少ないレビューにも、ユーザーに有益な情報を含んでいるレビューが存在している。スマホアプリは、度々アップデートを行っていくため、バージョンによってはレビューで指摘された部分に変化しており、情報が参考にならないことがある。また、アップデート後のレビューは比較的新しいため参考数が少なく、ユーザーは新しいレビューの中からどれが参考になるか判断しづらい。

本研究では、スマホアプリの比較的新しいレビューから、将来多くのユーザーに参考になると評価されるレビューを事前に発見する手法を提案する。

## 2. 関連研究

金ら[3]は、kakaku.com にある各社コーヒーマーカーのレビューに線形判別分析を用いて、参考数を集めやすい単語と集めにくい単語に分類した。分類結果として、レビューの内容に質的な差異や、企業別に参考数を集めるレビューに違いがあったとした。

コーヒーマーカーという同じジャンルでも、商品によるレビュー特徴が違っている。これにより、分類した参考数を集めやすい単語が違う商品では当てはまらないのが課題である。そのため、分類方法では、予測対象によって適切な特徴量を掴める手法の選択が必要である。この点を参考に、本研究は、各スマホアプリごとに予測モデルの作成を行うことにする。

## 3. 提案手法

本研究は、参考になりやすいスマホアプリレビューの予測手法を Google Play に存在するレビューを用いて分析する。本研究は、図 1 の流れに従って研究を行う。以下では研究の流れについて詳述する。

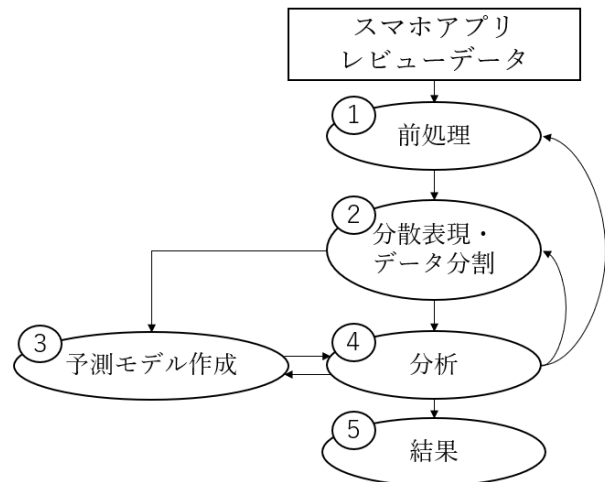


図 1 研究の流れ

## 3.1 レビューデータの前処理

はじめに、モデル作成時のノイズを取り除く。分散表現化時に影響を及ぼす  $n$  単語以下のレビューを、データから取り除く。これにより、分析に関係ない、または、結果に影響を及ぼす可能性があるデータを除くことができる。次に、レビュー文章の分かち書きを行い、文章の改行、スペースなどの文章に影響がない部分を除去し、レビュー文章を分散表現するための前処理を行う。また、30 日目に対象のレビューが参考になると予測するため、30 日目の値が 1 以上の数値を 1 と変換する。したがって、30 日目の参考数の値が 0 のレビューを参考にならないとして 0、1 以上のレビューを参考になったレビューとして 1 を、分類値と設定する。

## 3.2 レビュー文書分散表現・データ分割

レビュー文章の分散表現を行う際に CountVectorizer (以下、CountVec), TfidfVectorizer (以下、TF-IDFVec) の 2 つのアルゴリズムの中から選択する。

CountVec は、出現した単語をカウントし特徴量を求める分散表現手法である。殆ど出現しない単語を特徴量にしないために、最小単語数を決める必要がある。カウントした単語をそのまま特徴量とできるため、出現の多い単語の特徴量が以上に強くなる。

TF-IDFVec は、TF-IDF を用いた分散表現手法である。ある文章にしか現れないような単語の重要度を特徴量として表現することができるが、単語数が多い文書と少ない文書では、その文章に重要な単語であっても、特徴量への影響の大きさが変わる。

予測モデルの作成とテストが必要なため、レビューを予測モデル学習用データと、分析予測用テストデータに分割する。

<sup>†</sup> 神奈川大学大学院工学研究科

Graduate School of Engineering, Kanagawa University

<sup>‡</sup> 神奈川大学工学部経営工学科

Faculty of Engineering, Kanagawa University

### 3.3 予測モデル作成

予測モデルとして `scikit-learn` のベルヌーイモデルを利用する。予測モデル学習用データを用いて、レビュー文書の分散表現を説明変数、参考数の分類値を目的変数として、チューニングする。

### 3.4 分析

分析予測用テストデータを予測モデルで分類する。予測結果の正解率 (式(1))、精度 (式(2)) を元に、前処理時の単語数、分散表現方法および参考数予測モデルのハイパーパラメータなどをチューニングする。

$$\text{正解率} = \frac{\text{正解したレビュー数}}{\text{全レビュー数}} \quad (1)$$

$$\text{精度} = \frac{\text{参考になると予測し正解したレビュー数}}{\text{参考になると予測したレビュー数}} \quad (2)$$

### 3.5 結果

結果として、予測モデルの精度とスマホアプリに適した予測モデルを選択する。

## 4. 評価と考察

### 4.1 評価対象

35種類のスマホアプリの中から、2022年11月7日～2023年4月17日の期間に30日間連続して取得できたレビューデータが1000件以上となる22種類のスマホアプリのレビューデータを対象とした。

レビューデータの前処理では、形態素解析ツールとして `MeCab` を使用し、5単語以下のレビューを削減した。また、`CountVec` の最小単語数を3とし、学習用データとテストデータの比率は7:3とした。

### 4.2 評価結果

評価結果の一部を、表1および図2に示す。表1には、各アプリの参考になったレビューの割合および、各分散表現を利用したモデルの分析結果を示す。各アプリ名は、一部省略している。精度と比較する指標として、使用したデータに実際に含まれている参考になったレビューの割合を示す。この手法を利用して分類したアプリレビューを参照する際に、参考になったレビューを見つける確率が変化したことが分かる。

図2は、TF-IDFVecを用いた予測モデルの正解率および精度から `CountVec` を用いた予測モデルの正解率および精度を引いた値を示す。

表1 参考になったレビューの割合、正解率、精度

スマホ アプリ名	参考になった レビューの割合	TF-IDFVec		CountVec	
		正解率	精度	正解率	精度
youtube	0.7934	0.8203	0.8362	0.6974	0.9125
ポケモンgo	0.7970	0.7863	0.8094	0.6860	0.9079
Uber Eats	0.2673	0.7580	0.6905	0.7464	0.6034
プロセカ	0.3914	0.7217	0.7642	0.7183	0.7032
モンスト	0.8223	0.7957	0.8119	0.7788	0.9023
へっばん	0.7347	0.7538	0.8244	0.6336	0.9063
ツイッター	0.7741	0.8024	0.8629	0.6979	0.9108
TMTM	0.7733	0.7914	0.8082	0.7554	0.8933
ワンピース	0.2772	0.7568	0.5948	0.7484	0.5615

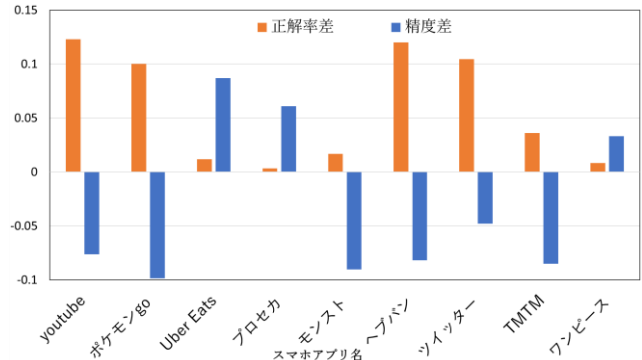


図2 TF-IDFVecから `CountVec` を引いた差

### 4.3 考察

表1から、Uber Eats およびワンピースが、両モデルでも比較的精度が低い。参考になったレビューの割合が低いため、モデルが特徴量を捉えきれない部分が多く、精度も低くなる。そのため、参考になったレビューの割合が少ないスマホアプリでは、精度が低くなる。しかし、参考になったレビューの割合と比べて精度の方が高いため、この手法を使用した方が、参考になりやすいレビューを見つける確率が高い。`CountVec` で作成した予測モデルでは、TF-IDFVecで作成したモデルと比較して、正解率と精度の差が大きいスマホアプリが複数存在している。`CountVec` で作成した予測モデルでは、精度が比較的高いが、正解率が低くなっているのが特徴である。これは、参考にならないと予測したレビューの正解率が精度より低いためである。

2つの結果を比較するために作成した図2では、TF-IDFVecで作成したモデルの方が、正解率が図2のアプリでは優れている。しかし、研究目的となるユーザーが実際に欲しい情報は、参考になると判断されたレビューのため、精度の方が重要だと考えられる。精度はスマホアプリによって、モデルにより優劣が変わっている。`CountVec` で作成した予測モデルの精度が比較的低いスマホアプリは、参考になったレビューの割合が低い傾向が出ている。そのため、精度を高めるためには、各スマホアプリレビューの特徴を掴んで、分散表現方法が正しい設定になるか見極める必要がある。

### 5. おわりに

本研究では、ユーザーから評価されるスマホアプリレビューの予測手法について検討し、ユーザーから参考になったと評価されるレビューを予測することが可能であり、スマホアプリによってモデルの調整が必要であることを示した。

今後の課題として、他のモデルや、分散表現方法を検討する必要がある。また、各スマホアプリのレビューの特徴をとらえ、その特徴に合わせたアルゴリズムが必要となる。例として、レビュー評価の星の数、投稿日時を特徴量に含めることを検討する必要がある。

### 参考文献

- [1] Google Play, <https://play.google.com/store/games?device=windows> (参照 2023-06-01)
- [2] Mobile App Ratings and Reviews: Where to Start and How to Win, <https://www.apptentive.com/> (参照 2022-09-05)
- [3] 金勝鎮, 多田侖, 勝又壮太郎, 評価付き自然言語データの定量分析—どのような消費者レビューが「参考になった」を集めるのか?—, 行動計量学, 47(2), 99-109 (2020)