Automatic Classification of Non-Functional Requirements from Augmented App User Reviews

Mengmeng Lu

Peng Liang

2017 21st International Conference on Evaluation and Assessment in Software Engineering(EASE'17), Pages 344-353

背景

- ユーザレビューは開発者がユーザの要求や不満を理解するのに役立つ
- 莫大な数のユーザレビューの中から有益な情報をとってくるのは大変
  - Facebookでは1日に4,275件のレビューが追加される（少ないものでも約23件）
- ユーザレビューを自動で分析する手法が提案されている[1,2,3]
  - バグレポート・機能要求・ユーザエクスペリエンス・レイトなどに分類[1]

[1]W. Maalej and H. Nabil. 2015. Bug report feature request or simply praise? On automatically classifying app reviews. In Proceedings of the 23rd IEEE International Requirements Engineering Conference (RE'15). IEEE, 116-125.
[2]S. Di Panichella, A. Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall. 2015. How can I improve my app? Classifying user reviews for software maintenance and evolution. In Proceedings of the 31st IEEE International Conference on Software Maintenance and Evolution (ICSME'15). IEEE, 281– 290.
[3]X. Gu and S. Kim. What parts of your apps are loved by users? 2015. In Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE'15). IEEE, 760-770.

アプリにおける非機能要件（NFR）

- NFRはユーザエクスペリエンスにおいて重要
  - NFRはソフトウェアシステムの品質の関する内容
  - アプリ開発過程できちんと認識すべき
- NFRはよく見落とされてしまう

- 既存研究ではユーザレビューをNFRに自動分類する手法は提案されていない

目的とアプローチ

- 目的
  - アプリ開発者のNFRへの認識を支援すること

- アプローチ
  - レビュー内容をNFRに自動分類する手法の構築
    - 4つの分類技術と3つの機械学習をそれぞれ組み合わせた手法を提案

提案手法

- 以下をそれぞれ組み合わせる
  - 分類技術
    - Bag of Words
    - TFIDF
    - Chi Squared
    - Augment User Reviews Bag Of Words（提案手法）
  - 機械学習アルゴリズム（既存手法で適用されていた）
    - ナイーブベイズ
    - J48
    - Bagging

Bag of Words(BOW)

- 情報検索において最もポピュラーな手法のひとつ
  - 単語の出現回数を特徴量とする

$$X_j = (x_{1,j} ... x_{i,j} ... x_{n,j})$$

X_j: レビュー j に含まれる単語のベクトル
$x_{i,j}$: レビュー j 中の単語 i の出現階数
n: 全てのレビューの単語数

R1：りんごとレモンとレモン

R2：りんごとみかん

R3：りんごとりんごとみかん

	りんご	レモン	みかん
R1	1	2	0
R2	1	0	1
R3	2	0	1

TFIDF

- 文d内の単語の出現頻度（TF）とある単語が出現する文章の頻度の逆数（IDF）をかけた値
  - どの文にも現れる単語は重要な単語ではない

$$TF-IDF(word_{i,j}) = f_{i,j} * \log \frac{total_user_reviews}{total_user_reviews_with_word_i}$$

$f_{i,j}$: レビュー j 中の単語 i の出現階数

R1：りんごとレモンとレモン

R2：りんごとみかん

R3：りんごとりんごとみかん

	りんご	レモン	みかん
R1	0.28	0.96	0
R2	0.61	0	0.78
R3	0.84	0	0.57

Chi Squared（カイ二乗値）

- 特徴選択アルゴリズムのひとつ
  - BoWの特徴から分類カテゴリに重要な特徴を選択
  - TFIDFの考え方を参考
  - 特定の分類カテゴリにのみ含まれる単語が重要

$$CHI^2(t_i, C_k) = \frac{N * (ad - bc)^2}{(a + c) * (b + d) * (a + b) * (c + d)}$$

Table 1: Variables of Formula (2)

	Belong to category C_k	Do not belong to category C_k
The number of user reviews with term t_i	a	b
The number of user reviews without term t_i	c	d

N: 学習するレビュー数

  - 分類カテゴリごとにCHIの大きい単語の上位N％を特徴単語とする

Nの求め方

- 二分探索法を用いる

```
begin = 1%
end = 100%
n = 100%
initialize best_classification_result according to n
while ( begin < end)
    n = ( begin + end ) / 2
    get classification_result_in_F-measure according to n
    if classification_result_in_F-measure < best_classification_result
        begin = n
    else
        end = n
        best_classification_result = classification_result_in_F-measure
n = end
return n
```

Algorithm 1: Binary search method for getting the value of *n*.

- nの値はNaïve Bayes, J48, Baggingでそれぞれ9%, 11%,17%の時が精度が最も良かった

Augment User Reviews Bag Of Words(AUR-BoW)

- 多くのレビューは100単語未満で非常に短く分類が難しい
- レビューに類似している単語を補完することで分類しやすくする

”crash every time ”

▼

”crash every time freezing log everytime ”

Augment User Reviews Bag Of Words(AUR-BoW)

- レビューと単語の類似度をWord2Vecを用いて算出
- 類似度の高い順に単語をN個補完する

$$sim(r_k, t_j) = \sum_{i=1}^{n} \left(w_{k,i} * sim(t_{k,i}, t_j) \right) \quad N = \theta * user_review_length$$

$sim(r_k, t_j)$：レビューr_kと単語t_jの類似度
$sim(t_{k,i}, t_j)$：レビューr_kに含まれる単語$t_{k,i}$と単語t_jの類似度
$w_{k,i}$：重み
θ：閾値（0~2までの値をとる）
n：レビューr_kに含まれる単語数
N：類義語として補完する単語数

- θの値はNaïve Bayes, J48,Baggingでそれぞれ1.9, 1.1,1.5のときが最も精度がよい

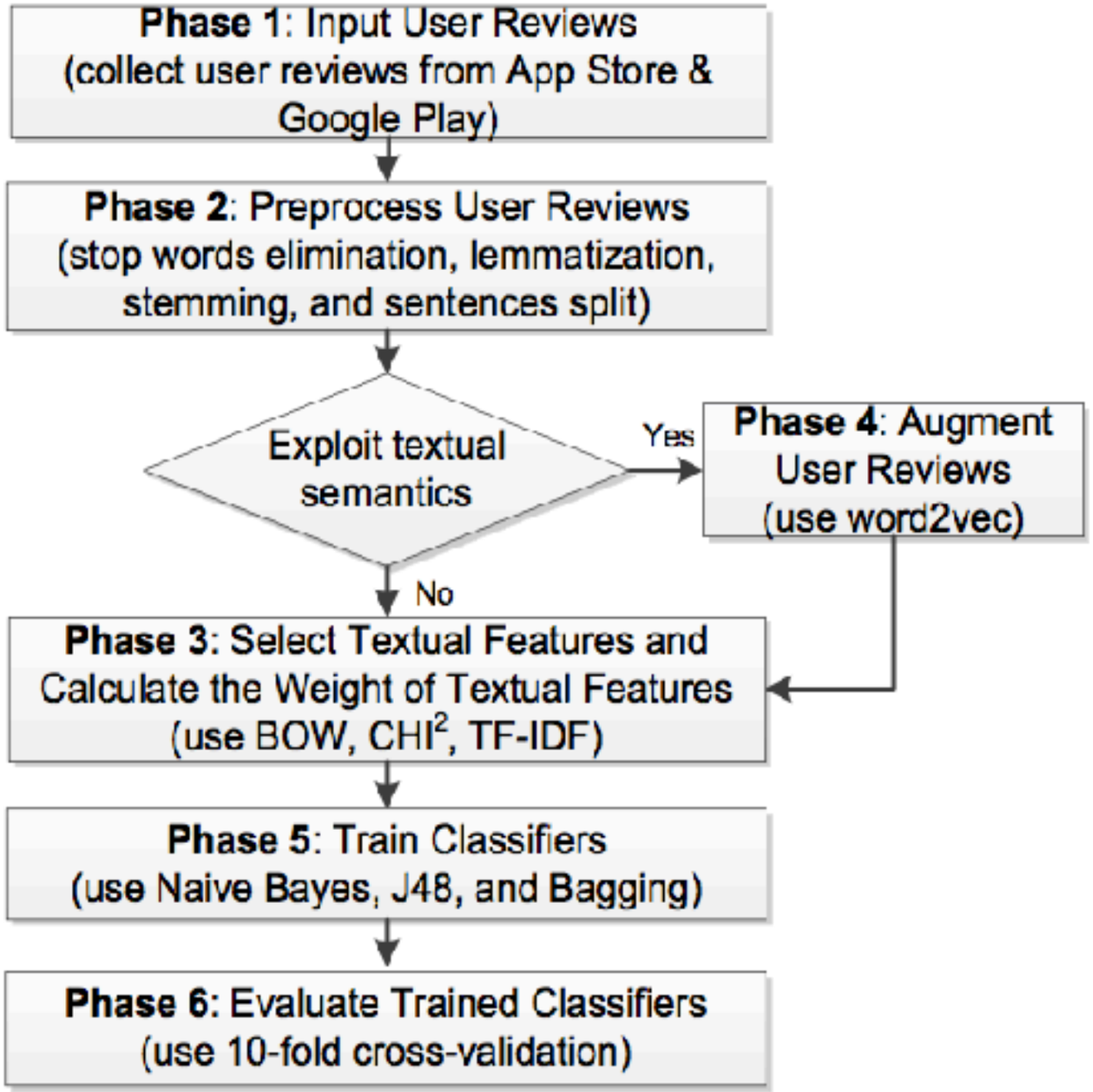
リサーチクエスチョン

RQ1：どの分類技術がユーザレビューを分類するのに適しているか

RQ2：どの機械学習アルゴリズムがユーザレビューを分類するのに適しているか

ユーザレビュー分類・評価手順

1. データの取得
2. 前処理
   - ストップワード除去
   - ステミング
   - 文章分割
3. 分類技術の選択
4. 3つの機械学習アルゴリズムを用いて学習
5. 評価実験
   - 10分割交差検証で評価



データセット

- iBooks(本のアプリ)とWhatapp(コミュニケーションアプリ)のユーザレビューの文4000件
- タイプ分類は著者らが手作業で実施

Type	#Sentence	Proportion
Usability	432	0.108
Reliability	587	0.147
Portability	119	0.029
Performance	121	0.030
FR	558	0.140
Others	2183	0.546
Total	4000	1.000

評価指標

- Precision，Recall，F値，Weighted averageを用いる
  - Weighted average(加重平均)：各ラベルごとの文章数を重みとして考慮して平均する

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{5}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$F - measure_i = \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i} \tag{7}$$

$$\begin{aligned} &Weighted\ average(Precision/Recall/F - measure) \\ &= \frac{\sum_{i \in type}(precision_i/recall_i/F - measure_i) * Number_i}{total_review_sentences} \end{aligned} \tag{8}$$

実験結果

Table 4: Results of the Combinations of One Classification Technique from BoW, TF-IDF, CHI^2, and AUR-BoW with One Machine Learning Algorithm from Naïve Bayes, J48, and Bagging

	BoW			TF-IDF			CHI^2			AUR-BoW		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Naïve Bayes	0.665	0.644	0.654	0.668	0.649	0.658	0.669	0.654	0.661	**0.720**	0.653	0.685
J48	0.656	0.674	0.665	0.661	0.678	0.670	0.661	0.680	0.670	0.677	0.690	0.684
Bagging	0.685	0.694	0.690	0.689	0.699	0.694	0.688	0.697	0.692	0.714	**0.723**	**0.718**

- AUR-Bow（提案手法）とBaggingアルゴリズムの組み合わせが最も精度がよい
  - F1値が最も高くバランスの取れた手法
  - RQ1の答え：AUR-BoW
  - RQ2の答え：Baggingアルゴリズム

分類の種類ごとの結果

Table 5: Percentages of Classified Sentences in the Dataset and Classification Results of Various Types Obtained by the Combination of AUR-BoW and Bagging

Type	Proportion	Precision	Recall	F1
Usability	0.108	0.757	0.565	0.647
Reliability	0.147	0.595	0.552	0.572
Portability	0.029	0.632	0.327	0.431
Performance	0.030	0.596	0.233	0.335
FR	0.140	0.630	0.587	0.608
Others	0.546	0.770	0.881	0.822
Weighted Average	1.000	0.714	0.723	0.718

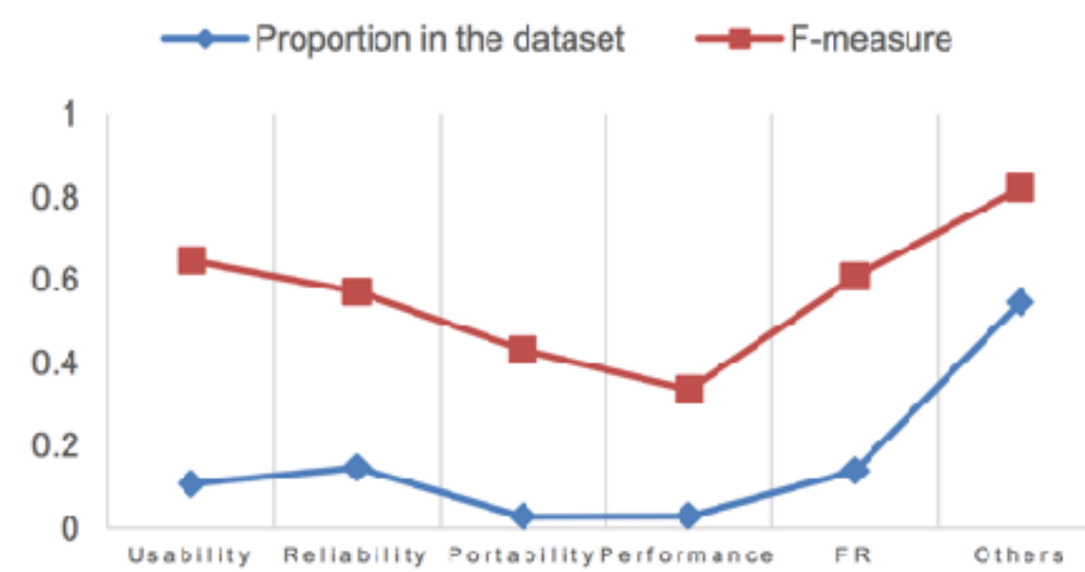


Figure 2: Comparison between the F-measure and proportion of each type in the dataset.

- UsabilityのF値がReliabilityとFRよりも高い
  - ReliabilityとFRは共通する単語が多かった
- 分類する種類によって精度が異なる（F1値：0.335~0.822）
  - データ数が少ないと精度が低い傾向がある
  - データ数を増やすことで精度向上を目指せる？

まとめと所感

- まとめ
  - アプリのレビューからNFRを自動抽出する手法を提案
  - 提案手法とBaggingアルゴリズムの組み合わせ手法が最も精度がよかった

- 所感
  - 単語を補完するという発想はおもしろかった
  - この論文では1文を分類しているが，応用したら複数の文でも分類可能そう
    - 複数の文を抽出するのが大変ですが・・・