Classifying/Categorizing App Store Reviews

Emma Gardner

Introduction

Motivation

Benefits of gathering, analyzing, and implementing user feedback[1]:

- Improving the user experience
- Improving the quality of the final product
- Retaining customers
- Reducing risk
- Greater agility

App store reviews provide an outlet for direct feedback from users to developers

Manually sorting through reviews is too time-consuming

Problem to Be Solved

Automatically classify reviews as actionable or unactionable and categorize them based on their contents

Technique and Implementation

Approach

Implement and Evaluate 4 Classifiers and 1 Categorizer Using Node.js (JavaScript Runtime Environment)

Generated Training Data and Gathered Testing Data

Training data: prompted ChatGPT[2] for unique actionable and unactionable reviews (1,000 each)

Removed duplicates → 681 actionable and 621 unactionable (1,302 total)

Testing data: gathered 100 actionable/unactionable reviews (200 total) of various applications on the Google Play, Apple App, and Microsoft Stores

Manually categorized the training and testing reviews into 11 categories:

App	Pricing	Updates/Versions	Model
GUI	Feature/Functionality	Resources	Company
Contents	Improvement	Security	

Manually Implemented the Naive Bayes Classifier[3]

Using the training data, computed the probability of…

1. Each word in the actionable reviews appearing in an actionable review
2. Each word in the unactionable reviews appearing in an unactionable review
3. A review being actionable
4. A review being unactionable

Using the real/test data…

5. For each review, computed the likelihood of it being actionable.
    a. If the review's probability is greater than 0.5, mark it as actionable.
    b. Else, mark it as unactionable.

Used brain.js' LSTM and GRU for Classification

Implemented LSTM (Long Short-Term Memory)[4] and GRU (Gated Recurrent Unit)[5] using methods included in brain.js

brain.js: GPU-accelerated library for neural networks[6]

- Used the *.train(trainingData, options)* method to train
  - Trained each network for 500 iterations (~8 hours)
- Ran on the real/test data using the *.run(input)* method to get the classification

Classification/Categorization with cfc-classifier

What is CFC (Class-Feature-Centroid)[7]?

- Machine learning algorithm for text categorization
- Based on a research paper by Hu Guan et al. titled “A class-feature-centroid classifier for text categorization”

Used cfc-classifier npm package[8] for classification (actionable/unactionable) and categorization (app, GUI, contents, etc.)

- Used the *.train()* method to train
- Ran using the *.classify(input)* method to get the classification/categorization

Evaluation

Classification Results

	Average of Methods	Naive Bayes	CFC	GRU	LSTM
Accuracy	59.75	59.5	66	56	57.5
Precision	61.59	67.27	61.76	63.04	54.29
Recall	61.25	37	84	29	95
F-Score	59.94	47.74	71.19	39.73	69.09

All results are in percentages.

Categorization Results

	Average of All Categories	App	GUI	Contents	Pricing	Feature/ Functionality
Accuracy	81.45	66.5	77	65	81.5	71
Precision	38.37	52.78	8.33	47.06	93.48	N/A
Recall	23.35	27.54	2.78	11.59	55.84	0
F-Score	31.42	36.19	4.17	18.6	69.92	N/A
	Improvement	**Security**	**Company**	**Updates/ Versions**	**Model**	**Resources**
Accuracy	82.5	93.5	94	87.5	89.5	88
Precision	55.56	0	44.44	37.04	25	20
Recall	13.89	0	36.36	55.56	20	33.33
F-Score	22.22	N/A	40	44.44	22.22	25

All results are in percentages.

Any questions?

Resources

1. Frank, Jeri. “Why User Feedback Is So Important For Software Development.” Forbes, Forbes Magazine, 1 June 2022, www.forbes.com/sites/forbesbusinesscouncil/2022/05/31/why-user-feedback-is-so-important-for-software-development/.

2. OpenAI. “ChatGPT.” Chat.openai.com, OpenAI, 2023, chat.openai.com/.

3. Lee, Jonathan. Notes on Naive Bayes Classifiers for Spam Filtering, 2018, https://courses.cs.washington.edu/courses/cse312/18sp/lectures/naive-bayes/naivebayesnotes.pdf.

4. Brownlee, Jason. “A Gentle Introduction to Long Short-Term Memory Networks by the Experts.” Machine Learning Mastery, 6 July 2021, https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

5. Chung, Junyoung, et al. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence ...” arXiv, 11 Dec. 2014, https://arxiv.org/pdf/1412.3555v1.pdf.

6. “brain.js.” npm, www.npmjs.com/package/brain.js. Accessed 26 Oct. 2023.

7. Guan, Hu, et al. “A Class-Feature-Centroid Classifier for Text Categorization: Proceedings of the 18th International Conference on World Wide Web.” ACM Conferences, 1 Apr. 2009, https://dl.acm.org/doi/abs/10.1145/1526709.1526737.

8. “cfc-classifier.” npm, www.npmjs.com/package/cfc-classifier. Accessed 26 Nov. 2023.