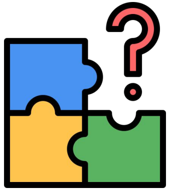
AUTOMATED USER REVIEW ANALYSIS TO FACILITATE POTENTIAL MOBILE APPLICATION EVOLUTION

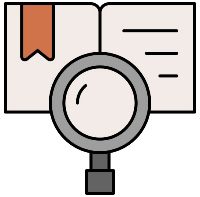
219337X-A.D.S.R.Gunathilaka
(Supervisor: Dr. Nisansa de Silva)

Content

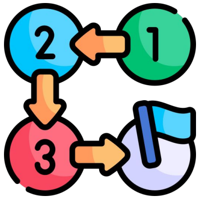
Introduction



Research Problem



Literature Survey



Research Phases

Summary

Introduction

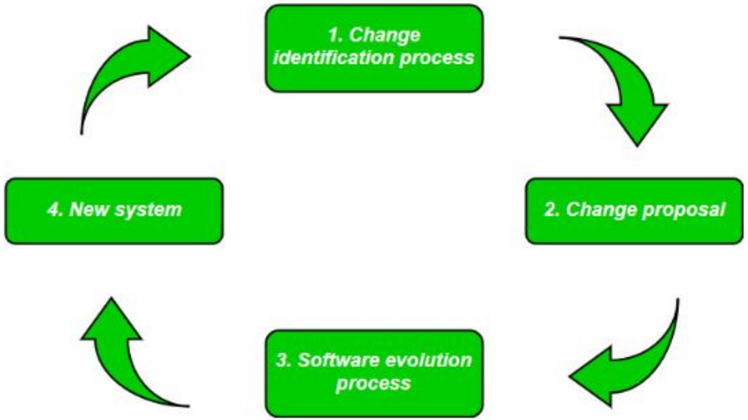
What is Software Evolution?

“Software evolution is the ongoing process of updating and improving software to keep up with changing needs, boost performance, and stay relevant. It ensures that the software keeps working properly, stays secure, and meets user expectations as circumstances and technology change.”

Why Software Evolves?

- External Drivers:
  - Changing user requirements and business needs
  - Market competition and technological advancements
  - Security threats and regulatory compliance

- Internal Drivers:
  - Bug fixes and performance optimization
  - Code maintainability improvements
  - Technical debt management



Software Evolution

Four Main Types

1. Corrective Evolution
   - Bug fixes, security patches, performance issue resolution
2. Adaptive Evolution
   - Platform updates, environment changes, new technology integration
3. Perfective Evolution
   - New features, user experience improvements, optimization
4. Preventive Evolution
   - Code restructuring, documentation updates, maintainability improvements

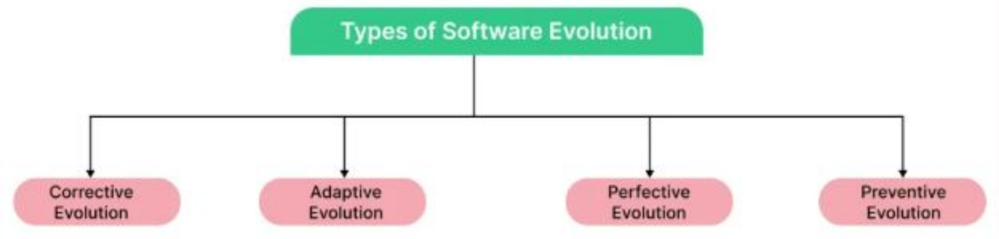


Fig 1.1 Types of Software Evolution

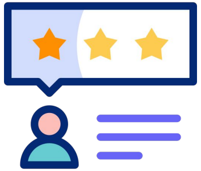
Importance of user feedback in the context of mobile app development

- User involvement is a major contributor to success of software projects [1].
- Feedback typically contains multiple topics related to the application such as user experience issues, bug reports, and feature requests [2][3].
- Most of the feedback given by the users after a new release and the frequency of feedback submitted decreases over the time [3].
- Feedback content has an impact on download numbers of the application.
- According to a study by W. Maalej [3] majority of low star rating feedback usually contains shortcomings and bug reports of the application where four to five star ratings mainly consist of praise.It was noted that the feature requests are mostly coming from three to five star rating feedback.
- User comments can be used to improve user satisfaction of software products [4].

1 M. Bano and D. Zowghi, “A systematic review on the relationship between user involvement and system success,” Information and Software Technology, vol. 58, 06 2014.
2 D. Pagano and B. Bruegge, “User involvement in software evolution practice: A case study,” 05 2013.
3 D. Pagano and W. Maalej, “User feedback in the appstore: An empirical study,” 07 2013.
4 H. Li, L. Zhang, L. Zhang, and J. Shen, “A user satisfaction analysis approach for software evolution,” 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 2, pp. 1093–1097, 2010.

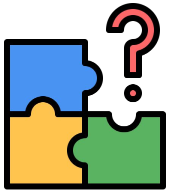
Introduction:Types of User feedback

- User feedback can be categorized into two types [5]:
  - Implicit feedback
  - explicit feedback



[5] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, “Toward data-driven requirements engineering,” IEEE Software, vol. 33, pp. 48–56, 01 2015.

Research Problem



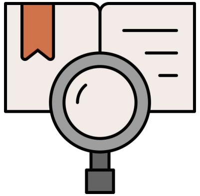
Research Problem

Despite the critical role of user reviews in mobile app evolution, developers face significant challenges in efficiently extracting actionable insights from the massive volume of unstructured feedback on app stores. While current NLP approaches have progressed from traditional machine learning to deep learning techniques, there remains a crucial need for:

1. More accurate and efficient methods to process large-scale user feedback
2. Better techniques to identify specific app aspects and associated user sentiments
3. Advanced solutions to automatically extract and classify user-reported issues and feature requests

This research addresses these challenges by exploring the potential of emerging NLP techniques, specifically ABSA and LLMs, to enhance the automated analysis of app reviews and streamline the mobile application evolution process.

Literature Survey



Traditional Machine Learning Approaches

Wiscom [14]

- Three-level analysis:
  - Meso: LDA for user complaints analysis
  - Micro: Linear Regression for text-rating inconsistency
  - Macro: Global marketplace trends
- First to use time-series on reviews

App Review Miner [15]

- Comprehensive analytics using LDA
- EMNB classifier for filtering non-informative reviews
- Topic modeling for grouping reviews
- Ranking scheme for prioritization

[14] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, “Why people hate your app: Making sense of user feedback in a mobile app store,” Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 08 2013.
[15] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, “Ar-miner: mining informative reviews for developers from mobile app marketplace,” in Proceedings of the 36th international conference on software engineering, 2014, pp. 767– 778.

Traditional Machine Learning Approaches

Anchiêta and Moura [17]

- Extended Chen et al.'s approach
- Evaluated different clustering techniques
- Focus on Brazilian Portuguese reviews

MARK Framework [8]

- Keyword-based semi-automated approach
- Analyst-driven analytical process
- Features:
    - Reviews filtered by keywords
    - Trend detection over time
    - Sudden change detection for issues

[17] R. T. Anchiêta and R. S. Moura, “Exploring unsupervised learning towards extractive summarization of user reviews,” in Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web, 2017, pp. 217–220.
[8] P. M. Vu, T. T. Nguyen, H. V. Pham, and T. T. Nguyen, “Mining user opinions in mobile app reviews: A keyword-based approach,” arXiv preprint arXiv:1505.04657, 2015.

Traditional Machine Learning Approaches

SUR-Miner [7]

- First pattern-based parsing approach
- Uses predefined sentence patterns
- Focus on structure and semantics
- Features:
  - Five-category classification
  - Direct aspect-opinion extraction
  - Interactive visualization diagrams

Guzman et al. [16]:

- Proposed taxonomy for app review classification
- Compared multiple algorithms:
  - Naive Bayes, SVM
  - Logistic Regression
  - Neural Networks
- Finding: Ensemble methods performed better

[7] X. Gu and S. Kim, “"what parts of your apps are loved by users?" (t),” 11 2015, pp. 760–770.
[16] E. Guzman, M. El-Haliby, and B. Bruegge, “Ensemble methods for app review classification: An approach for software evolution (n),” 11 2015, pp. 771–776.

Traditional Machine Learning Approaches

Maalej et al. [10]:

- Four-type classification system
- Combined multiple techniques:
  - Text classification
  - Natural language processing
  - Sentiment analysis
- Results: 88-92% precision, 90-99% recall

Dhinakaran et al. [19]:

- Active learning to reduce labeling effort
- Three uncertainty sampling strategies
- Applied to 4400 app reviews

[10] W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, “On the automatic classification of app reviews,” Requirements Engineering, vol. 21, 09 2016.
[19] V. Dhinakaran, R. Pulle, N. Ajmeri, and P. Murukannaiah, “App review analysis via active learning: Reducing supervision effort without compromising classification accuracy,” 08 2018, pp. 170–181.

Traditional Machine Learning Approaches

Guo and Singh [22]:

- Caspar: Action-problem pair extraction
- Focus on mini stories from reviews
- Specific suggestions for developers

Deep Learning Approaches

Stanik et al. [20]:

- Deep Convolutional Neural Network
- Embedding layer with word2vec/FastText
- English and Italian language support
- Finding: Comparable to traditional ML with domain expertise

Aslam et al. [21]:

- Combined textual and non-textual data
- Features:
  - Review counts
  - Submission rates
  - Review metadata
- Multi-class classifier

[20] C. Stanik, M. Haering, and W. Maalej, “Classifying multilingual user feedback using traditional machine learning and deep learning,” 09 2019, pp. 220–226.
[21] N. Aslam, A. RAMAY, X. KEWEN, and N. Sarwar, “Convolutional neural network-based classification of app reviews,” IEEE Access, vol. 8, pp. 1–11, 10 2020.

Deep Learning Approaches

Hadi and Fard [22]:

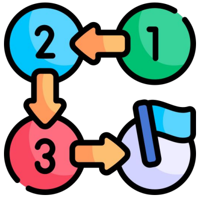
- Empirical study on six datasets
- Multiple classification settings

Henao et al. [23]:

- Monolingual vs multilingual BERT
- Key finding: Heavyweight transfer learning not always better

[22] M. A. Hadi and F. H. Fard, “Evaluating pre-trained models for user feedback analysis in software engineering: A study on classification of app-reviews,” 2021.
[23] P. Restrepo, J. Fischbach, D. Spies, J. Frattini, and A. Vogelsang, “Transfer learning for mining feature requests and bug reports from tweets and app store reviews,” 08 2021

Research Phases



Phase 1:
Aspect Based Sentiment Analysis On App Reviews

Introduction : Why ABSA?

“UI is awesome and easy to use but applications drains the battery faster.”

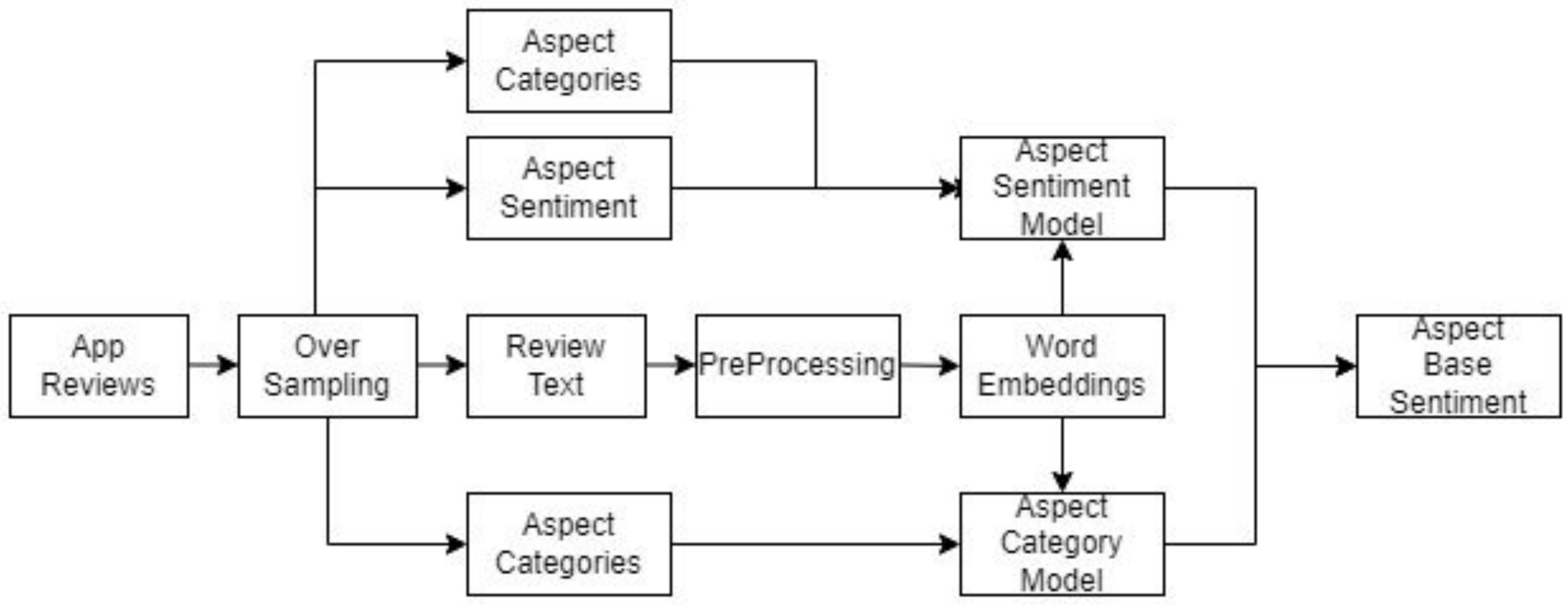
- Having the aspect information along with their respective sentiment leads to a fine-grained analysis [6].
- To support such analysis, we can utilize Aspect-Based Sentiment Analysis (ABSA) [7], which identifies the sentiment with respect to a specific aspect.
- Work done by N. Alturaief [8] et al is the first study that investigated the applicability of supervised ABSA to incorporate user feedback into requirement elicitation process.

- ABSA consists of three sub-tasks:

    - Aspect category classification.
    - Aspect term extraction.
    - Aspect sentiment analysis.

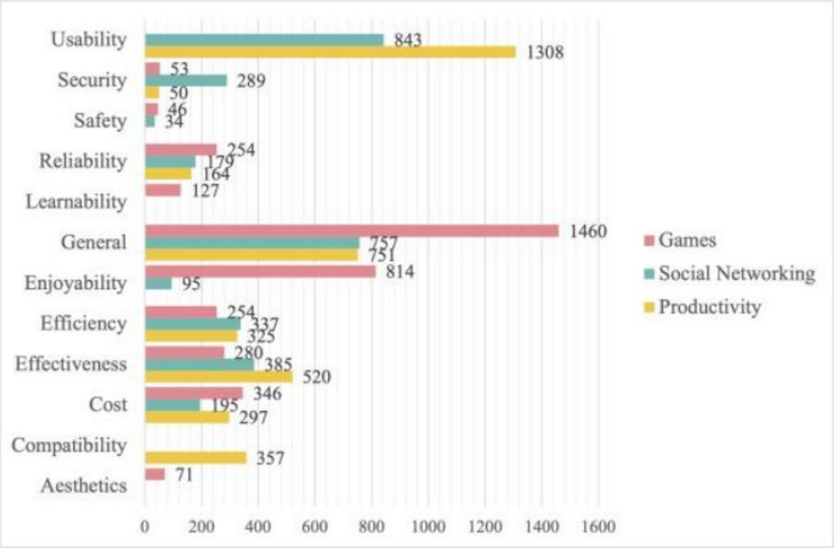
[6] Y. Li, B. Jia, Y. Guo, and X. Chen, “Mining user reviews for mobile app comparisons,” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 3, pp. 1–15, 017.
[7] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 168–177.
[8] N. Alturaief, H. Aljamaan and M. Baslyman, ”AWARE: Aspect-Based Sentiment Analysis Dataset of Apps Reviews for Requirements Elicitation,” 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 2021, pp. 211-218, doi: 10.1109/ASEW52652.2021.00049

Methodology : Proposed Approach Overview



Methodology : Dataset [8]

- **AWARE** is benchmark dataset of **11,323** apps reviews that are annotated with aspect terms,categories, and sentiment.
- It contains reviews that were collected from three domains: **productivity**, **social networking**, and **games**.
- The data set contains two aspect definitions
  - **Aspect Term**: A term describing an aspect of an app that is expressed by the sentiment and that exists in the sentence.
  - **Aspect Category**: A predefined set of domain-specific categories.



[8] N. Alturaief, H. Aljamaan and M. Baslyman, "AWARE: Aspect-Based Sentiment Analysis Dataset of Apps Reviews for Requirements Elicitation," 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 2021, pp. 211-218, doi: 10.1109/ASEW52652.2021.00049

Methodology : OverSampling the Data

- Contextual augmentation by **Google Bert** [9].
    - Contextual words embeddings assigns each words a representation based on its context. We used substitute actions for augmenting data. In substitute, length of sentence is same but some words are replaced. We utilized the NLPAug [10] open source python package for data augmentation.

- Data Augmentation by Round-trip translation (**RTT**).
    - Round-trip translation (RTT) is additionally referred to as recursive, back-and forth, and bi-directional translation. it's the method of translating a word, phrase or text into another language (forward translation), then translating the results back to the first language (back translation) .RTT is used as augmentation technique to extend the training data. We used Roundtrip translation python package [11] to augment data.

[9] Kobayashi, Sosuke. (2018). Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. 452-457. 10.18653/v1/N18-2072.
[10] https://github.com/samhavens/roundtrip.
[11] https://github.com/makcedward/nlpaug

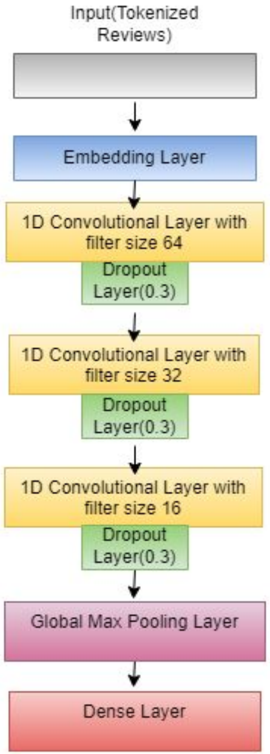
Methodology : Preprocessing

Methodology : Embeddings

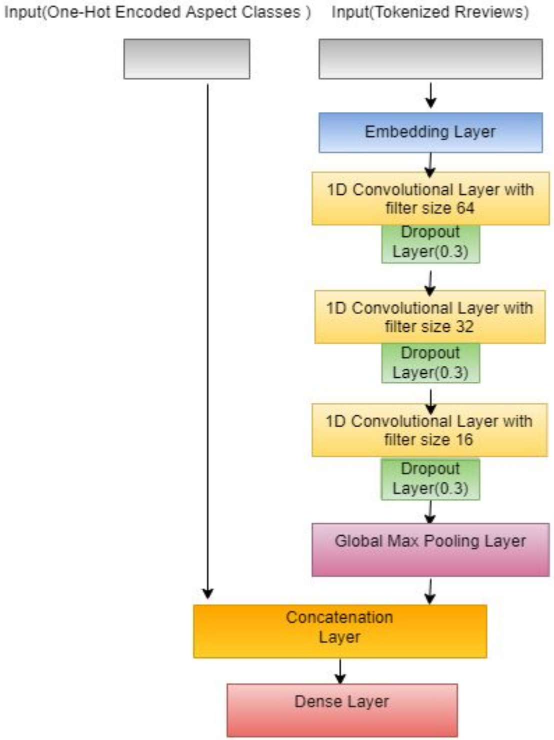
Pre-trained Models:

- **FastText:** Wiki-news model, with 1 million word vectors and 300 dimensions, trained on Wikipedia 2017, UMBC web based corpus and statmt.org news dataset
- **Glove:** Pre-trained model, trained trained on Wikipedia data with 6 billion tokens, 100 dimensions and a 400,000-word vocabulary.
- **Word2Vec:** Google word2vec model, trained on Google news data (about 100 billion words); it contains 3 million words and phrases and was fit using 300-dimensional word vectors.

Methodology : Feature extraction and classification



(a) Aspect Category Classification Model



(b) Aspect Sentiment Classification Model

Experiments & Results: Aspect Category Classification

Dataset	Word Embedding	Preprocessing	BERT	RTT(DE)	RTT(CN)	RTT(TR)	RTT(JP)
Productivity	Fasttext	Disabled	0.60	0.59	0.25	0.61	0.60
		Enabled	0.63	0.61	0.23	0.62	0.59
	Word2Vec	Disabled	0.61	0.62	0.24	0.61	0.61
		Enabled	0.62	0.62	0.26	0.61	0.60
	Glove	Disabled	0.54	0.53	0.24	0.52	0.55
		Enabled	0.56	0.57	0.25	0.58	0.58
Gaming	Fasttext	Disabled	0.42	0.45	0.19	0.35	0.43
		Enabled	0.40	0.39	0.22	0.28	0.45
	Word2Vec	Disabled	0.42	0.41	0.23	0.37	0.44
		Enabled	0.39	0.42	0.21	0.37	0.44
	Glove	Disabled	0.42	0.44	0.20	0.34	0.42
		Enabled	0.30	0.30	0.21	0.24	0.31
Social	Fasttext	Disabled	0.62	0.62	0.58	0.25	0.60
		Enabled	0.60	0.61	0.58	0.27	0.60
	Word2Vec	Disabled	0.60	0.62	0.61	0.29	0.61
		Enabled	0.58	0.62	0.61	0.28	0.61
	Glove	Disabled	0.54	0.56	0.54	0.27	0.55
		Enabled	0.54	0.55	0.55	0.26	0.57
Average	Fasttext	Disabled	0.55	0.56	0.34	0.41	0.55
		Enabled	0.55	0.54	0.35	0.39	0.55
	Word2Vec	Disabled	0.55	0.55	0.36	0.43	0.56
		Enabled	0.53	0.56	0.36	0.42	0.55
	Glove	Disabled	0.50	0.51	0.33	0.38	0.51
		Enabled	0.47	0.48	0.34	0.36	0.49

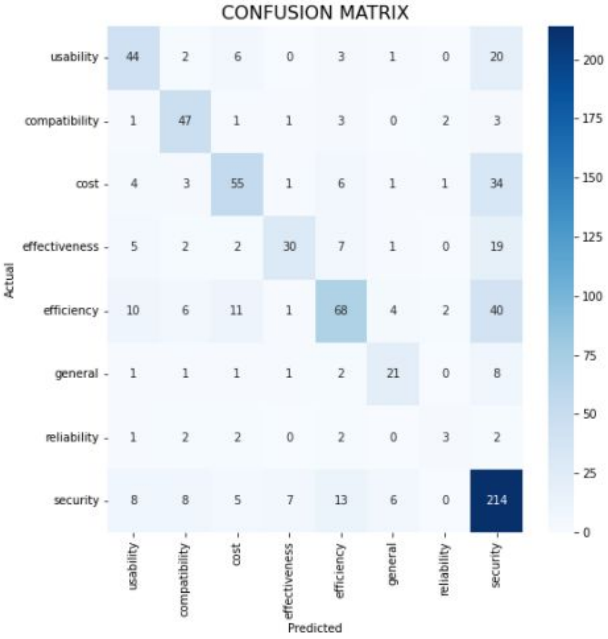
Experiments & Results: Aspect Sentiment Classification

Dataset	Word Embedding	Preprocessing	BERT	RTT(DE)	RTT(CN)	RTT(TR)	RTT(JP)
Productivity	Fasttext	Disabled	0.81	0.81	0.62	0.80	0.78
		Enabled	0.79	0.79	0.63	0.81	0.81
	Word2Vec	Disabled	0.80	0.80	0.62	0.82	0.80
		Enabled	0.79	0.79	0.64	0.82	0.81
	Glove	Disabled	0.80	0.79	0.61	0.79	0.81
		Enabled	0.79	0.80	0.62	0.80	0.77
Gaming	Fasttext	Disabled	0.70	0.71	0.68	0.65	0.70
		Enabled	0.71	0.70	0.68	0.65	0.71
	Word2Vec	Disabled	0.70	0.70	0.67	0.64	0.70
		Enabled	0.72	0.69	0.68	0.66	0.70
	Glove	Disabled	0.71	0.72	0.70	0.65	0.70
		Enabled	0.70	0.69	0.69	0.65	0.70
Social	Fasttext	Disabled	0.83	0.80	0.81	0.63	0.83
		Enabled	0.82	0.83	0.81	0.62	0.82
	Word2Vec	Disabled	0.81	0.86	0.81	0.64	0.80
		Enabled	0.82	0.86	0.82	0.64	0.83
	Glove	Disabled	0.82	0.84	0.82	0.64	0.81
		Enabled	0.79	0.85	0.82	0.63	0.81
Average	Fasttext	Disabled	0.78	0.78	0.71	0.70	0.77
		Enabled	0.78	0.78	0.71	0.70	0.78
	Word2Vec	Disabled	0.77	0.79	0.70	0.70	0.77
		Enabled	0.78	0.78	0.72	0.71	0.78
	Glove	Disabled	0.78	0.79	0.71	0.70	0.78
		Enabled	0.76	0.78	0.71	0.70	0.76

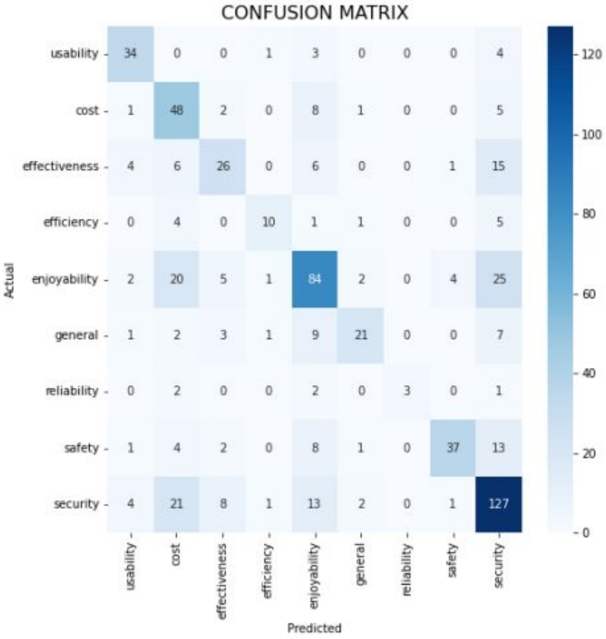
Experiments & Results: Summery

Task		Baseline	Results	Metric
Aspect Category Classification	Productivity	0.33	0.62	F1
	Social Networking	0.32	0.62	F1
	Games	0.32	0.42	F1
Aspect Sentiment Classification	Productivity	68.71%	80%	Acc.
	Social Networking	69.72%	86%	Acc.
	Games	67.49%	70%	Acc.

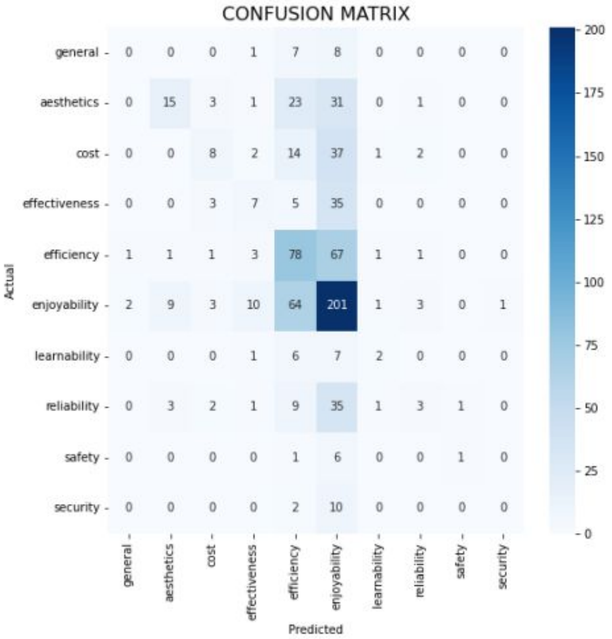
Experiments & Results: Error Analysis



(a) Productivity



(b) Social Networking



(c) Game

Conclusion

- The results showed that our approach could archive F1 scores of **0.62**, **0.42**, and **0.62** in the aspect category classification task, and accuracy of **0.80**, **0.70**, and **0.86** for the aspect sentiment classification task in **Productivity**, **Game**, and **Social Networking** domains respectively.
- As a future work we intend to investigate the possibility of using transformer based models to improve the results further.

Publication

- Aspect-based Sentiment Analysis on Mobile Application Reviews.
    - This paper was published in the 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer), where we introduced a novel CNN-based approach for analyzing mobile app reviews using Aspect-Based Sentiment Analysis [39].

Phase 2: Automatic Analysis Of App Reviews Using LLMs

Introduction : LLMs for App Review Analysis

Motivation:

- Commercial & open-source LLMs show promise for app review classification
- Potential for automated high-quality dataset creation
- Need for cost-effective solutions

Research Focus:

- Evaluating LLMs in zero-shot settings
- Using LLMs as autonomous annotators
- Fine-tuning open-source models
- Analyzing parameter impacts (Temperature, `Top_p`, Epochs, and Training Data Sample Size)

Key Questions:

- How do commercial LLMs perform in zero-shot classification?
- Can LLMs create reliable training datasets?
- How do fine-tuned open-source models compare?

LLMs As a Annotator

Wang et al. [28]:

- One of first studies on GPT-3 for annotation
- Augmented manually labeled data with GPT-3 pseudo-labels
- Improved model performance with constrained budgets
- Limitation: Quality still lagged behind human annotations

He et al. [29] - "Explain-then-annotate":

- Two-step approach:
  - Generate explanations using GPT-3.5
  - Construct chain-of-thought prompts
- Outperformed zero-shot and few-shot annotation

Zhang et al. [30] - LLAMA Framework:

- Combines active learning with prompt engineering
- Focus: Named entity recognition and relation extraction
- Result: Models outperformed teacher LLMs within hundreds of samples

[28] S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, “Want to reduce labeling cost? gpt-3 can help,” arXiv preprint arXiv:2108.13487, 2021.
[29] X. He, Z. Lin, Y. Gong, A. Jin, H. Zhang, C. Lin, J. Jiao, S. M. Yiu, N. Duan, W. Chen et al., “Annollm: Making large language models to be better crowd sourced annotators,” arXiv preprint arXiv:2303.16854, 2023.
[30] R. Zhang, Y. Li, Y. Ma, M. Zhou, and L. Zou, “Llmaaa: Making large language models as active annotators,” arXiv preprint arXiv:2310.19596, 2023.

LLMs As a Annotator(Continue..)

Zhou et al. [31]:

- Combined approach:
  - BERT for classification
  - CRFs for attribute value extraction
  - LLMs for data annotation
- Improved attribute recognition from customer queries

He et al. [32]:

- Integration with crowdsourced annotation
- Key finding: Task-specific models can outperform teacher LLMs
- Emphasis: Importance of maintaining human involvement

Tang et al. [34] - PDF Annotator:

- Human-LLM collaborative tool
- Focus: Multi-modal data from PDF catalogs
- Combines LLM capabilities with human guidance

[31] J. Zhou, W. Du, M. O. F. Rokon, Z. Wang, J. Xu, I. Shah, K.-c. Lee, and M. Wen, "Enhanced e-commerce attribute extraction: Innovating with decorative relation correction and llama 2.0-based annotation," arXiv preprint arXiv:2312.06684, 2023.
[32] Z. He, C.-Y. Huang, C.-K. C. Ding, S. Rohatgi, and T.-H. K. Huang, "If in a crowdsourced data annotation pipeline, a gpt-4," in Proceedings of the CHI Con ference on Human Factors in Computing Systems, 2024, pp. 1–25.
[34] Y. Tang, C.-M. Chang, and X. Yang, "Pdf Annotator: A human-llm collaborative multi-modal data annotation tool for pdf-format catalogs," in Proceedings of the 29th International Conference on Intelligent User Interfaces, 2024, pp. 419–430.

LLMs As a Annotator(Continue..)

Wang et al. [33]:

- Study: LLMs replacing human participants
- Critical limitations identified:
  - Misportrayal of marginalized groups
  - Flattening of group diversity

Yu et al. [35]:

- Focus: Corpus-based pragmatics
- Study of apology annotation
- Compared GPT-3.5 and GPT-4 with human annotators

Imamovic et al. [37]:

- Used ChatGPT for Appraisal Theory annotation
- Results:
  - High precision in detecting evaluative meaning
  - Low recall
  - Need for human oversight emphasized

[33] A. Wang, J. Morgenstern, and J. P. Dickerson, “Large language models cannot replace human participants because they cannot portray identity groups,” arXiv preprint arXiv:2402.01908, 2024.
[35] D. Yu, L. Li, H. Su, and M. Fuoli, “Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apology,” International Journal of Corpus Linguistics, 2024.
[37] M.Imamovic, S. Deilen, D. Glynn, and E. Lapshinova-Koltunski, “Using chatgpt for annotation of attitude within the appraisal theory: Lessons learned,” in Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII), 2024, pp. 112–123.

LLMs As a Annotator(Continue..)

Pangakis et al. [36]:

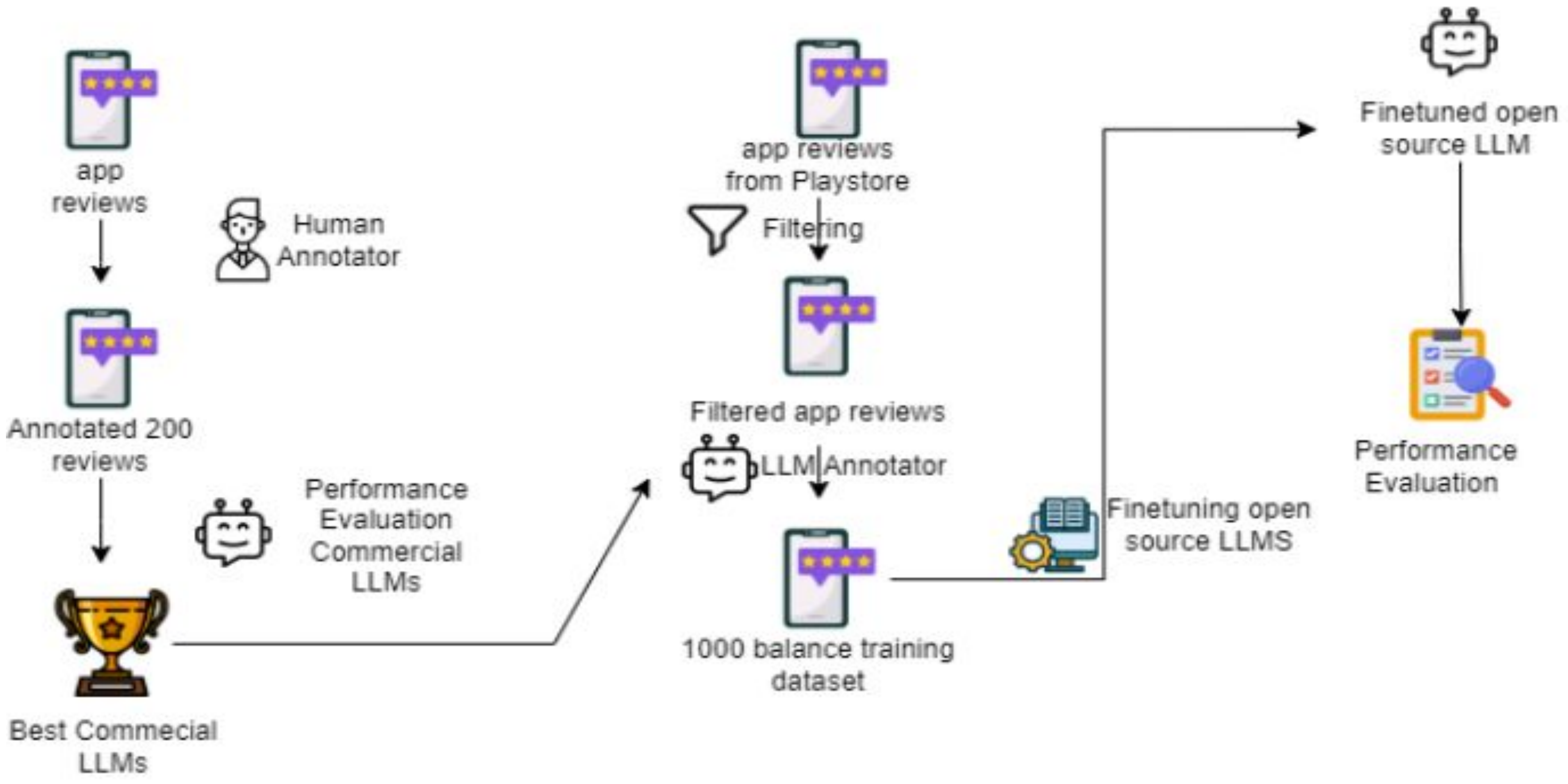
- Focus: Generative AI for automated annotation
- Proposed workflow:
  - Harness LLM potential
  - Ensure accuracy through human oversight

Tan et al. [38]:

- Synthesized recent advancements
- Covered:
  - Challenges
  - Future directions
  - Cross-domain applications

[36] N.Pangakis, S. Wolken, and N. Fasching, “Automated annotation with generative ai requires validation,” arXiv preprint arXiv:2306.00176, 2023.
[38] J. Tan, A. Zhang, X. Zhang, C. Xiao, Z. Ding, Y. Peng, C. Wu, X. Zhu, J. Zhou, and X. Huang, “Large language models for data annotation: A survey,” arXiv preprint arXiv:2402.13446, 2024.

Methodology : Proposed Approach Overview



Methodology : LLM Selection for Automated Annotation

Models selected for automated annotation:

- OpenAI's GPT-3.5 (gpt-3.5-turbo-0125) [5]
- Google's Gemini Pro 1.0 [6]

Selection rationale:

- Balance between cost-effectiveness and performance
- Compared to more advanced models (GPT-4 and Gemini Pro 1.5)

[5] https://platform.openai.com/docs/models/gpt-3-5-turbo
[6] https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/gemini-pro

Methodology : LLM Selection for Automated Annotation

Annotation approach:

- Zero-shot setting used
- Minimizes context size and costs
- Utilized annotation prompt specifically crafted to annotation process.
- Interacted via respective API endpoints

Methodology : LLM Selection for Custom Models

Hardware constraints:

- Single RTX 4090 GPU with 24GB VRAM

Selection criteria:

- Instruction-following capabilities
- JSON response formatting

Methodology : LLM Selection for Custom Models

Models selected:

- Llama-2-7b-chat-hf [7]
- Mistral-7B-Instruct [8]
- Falcon 7B Instruct [9]

[7] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
[9] https://huggingface.co/tiiuae/falcon-7b-instruct

Methodology : Benchmarking Dataset

The Benchmarking Dataset is a carefully curated subset of app reviews derived from Maalej and Nabil's 2015 study. Key features include:

1. 200 reviews in total, evenly distributed with 50 reviews per category.
2. Categories: Four main classes, slightly modified for LLM readability:
    1. Bug Reports
    2. Feature Requests
    3. User Experience
    4. Ratings

Methodology : Benchmarking Dataset

Categories and Definitions:

- **Bug Reports:** User comments identifying app issues such as crashes, incorrect behavior, or performance problems. These highlight functional problems requiring corrective action.

- **Feature Requests:** User suggestions for new features or enhancements in future updates. These include requests for features from other apps, content additions, or ideas to modify existing features.

- **User Experience:** Detailed narratives focusing on specific app features and their real-world effectiveness. These offer insights into usability, functionality, and overall satisfaction, often serving as informal documentation of user needs and app performance.

- **Ratings:** Brief textual comments reflecting the app's numeric star rating, primarily indicating overall user satisfaction or dissatisfaction without detailed justification.

Methodology : Dataset for Fine-Tuning Custom LLMs

Data Collection:

- **Source:** Google App Store
- **Total reviews collected:** 92,354
- Popular US applications ranked by `appfigures.com` [10]
- Over 90 distinct mobile applications

Selection Criteria:

- **Language:** English only filtered using `langdetect` Python library [11]
- **Initial selection:** 85,852 reviews (>10 words)
- **Additional selection:** 6,502 reviews (2-10 words)

[10] https://appfigures.com/top-apps/ios-app-store/united-states/iphone/top-overall
[11] https://pypi.org/project/langdetect/

Methodology : Dataset for Fine-Tuning Custom LLMs

Annotation Process:

- LLM: Open AI's GPT 3.5(according to our experiment results)
- Configuration:
  - Temperature: 1
  - Top_p value: 0.25
- Used annotation prompt template

Final Dataset:

- Total size: 10,000 reviews
- Distribution: 2,500 reviews per category (4 categories)

Methodology : Prompts for Annotation

Key Features:

- Boolean questions for each class
- Explanations required for each decision
- Designed for multi-category reviews

Structure:

- Series of questions and explanations
- Post-classification precedence order applied (bugs>feature>user experience>rating)

Methodology : Prompts for Annotation

Purpose:

- Enhance classification accuracy
- Capture nuanced, multi-faceted reviews

Rationale:

- Avoids oversimplification of complex reviews
- Encourages comprehensive consideration of all categories

Methodology : Prompts for Annotation

```
Task Description:

Review user reviews for mobile applications based on their content, sentiment, and ratings. Utilize the definitions provided to
classify each review into the appropriate category.

Definitions for Classification:

Bug Reports:
Definition: Bug reports are user comments that identify issues with the app, such as crashes, incorrect behavior, or performance
problems. These reviews specifically highlight problems that affect the app's functionality and suggest a need for corrective action.

Feature Requests:
Definition: Feature requests are suggestions by users for new features or enhancements in future app updates. These can include
requests for features seen in other apps, additions to content, or ideas to modify existing features to enhance user interaction
and satisfaction.

User Experience:
Definition: User experience reviews provide detailed narratives focusing on specific app features and their effectiveness in real
scenarios. They offer insights into the app's usability, functionality, and overall satisfaction, often serving as informal documentation
of user needs and app performance.
Differentiating Tip: Prioritize reviews that give detailed explanations of the app's features and their practical impact on the user.

Ratings:
Definition: Ratings are brief textual comments that reflect the app's numeric star rating, primarily indicating overall user satisfaction
or dissatisfaction. These reviews are succinct, focusing on expressing a general sentiment without detailed justification.
Differentiating Tip: Focus on reviews that lack detailed discussion of specific features or user experiences, and instead provide general
expressions of approval or disapproval.

Questions:

Q1.Does it sound like a Bug Report?: <True or False>
Q2.Explain why Q1 is True/False: <explanation>
Q3.Does it sound like a missing Feature?": <True or False>
Q4.Explain why Q3 is True/False: <explanation>
Q5.Does it sound like a User Experience?: <True or False>
Q6.Explain why Q5 is True/False: <explanation>
Q7.Does it sound like a Rating?: <True or False>
Q8.Explain why Q7 is True/False: <explanation>

Instructions to the Language Model:

Review Processing: Carefully read the provided app review and its star rating and answer all questions.

Output Format: Provide the classification results in the following JSON format:
{{
    "Q1.Does it sound like a Bug Report?": "<True or False>",
    "Q2.Explain why Q1 is True/False": "<explanation>",
    "Q3.Does it sound like a missing Feature?": "<True or False>",
    "Q4.Explain why Q3 is True/False": "<explanation>",
    "Q5.Does it sound like a User Experience?": "<True or False>",
    "Q6.Explain why Q5 is True/False": "<explanation>",
    "Q7.Does it sound like a Rating?": "<True or False>",
    "Q8.Explain why Q7 is True/False": "<explanation>"
}}

Review and Star Rating to Classify:
```

Methodology : Prompts for Fine-Tuning Custom Model

Key Components:

- Two primary templates:
  1. Task description and label definitions
  2. Explain-then-annotate pattern [15]

Fine-Tuning Approach:

- Maximum sequence length: 800 tokens
- **Output format:** JSON
- **Tool:** Hugging Face `SFTTrainer` Library

[15]] Xingwei He, Zhenghao Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. arXiv preprint arXiv:2303.16854.

Methodology : Prompts for Fine-Tuning Custom Model

Task Description:

Review user reviews for mobile applications based on their content, sentiment, and ratings. Utilize the definitions provided to classify each review into the appropriate category.

Definitions for Classification:

Bug Reports:
Definition: Bug reports are user comments that identify issues with the app, such as crashes, incorrect behavior, or performance problems. These reviews specifically highlight problems that affect the app's functionality and suggest a need for corrective action.

Feature Requests:
Definition: Feature requests are suggestions by users for new features or enhancements in future app updates. These can include requests for features seen in other apps, additions to content, or ideas to modify existing features to enhance user interaction and satisfaction.

User Experience:
Definition: User experience reviews provide detailed narratives focusing on specific app features and their effectiveness in real scenarios. They offer insights into the app's usability, functionality, and overall satisfaction, often serving as informal documentation of user needs and app performance.
Differentiating Tip: Prioritize reviews that give detailed explanations of the app's features and their practical impact on the user.

Ratings:
Definition: Ratings are brief textual comments that reflect the app's numeric star rating, primarily indicating overall user satisfaction or dissatisfaction. These reviews are succinct, focusing on expressing a general sentiment without detailed justification.
Differentiating Tip: Focus on reviews that lack detailed discussion of specific features or user experiences, and instead provide general expressions of approval or disapproval.

Instructions to the Language Model:

Review Processing: Carefully read the provided app review and its star rating and Classify the review into one of the following categories: "Bug", "Feature", "UserExperience", or "Rating".

Output Format: Provide the classification results in the following JSON format:

```
{{
    "Class": "<predition>"
}}
```

Review and Star Rating to Classify:
User Review : "Absolutely handy for those pics you don't need everyone else to see."
User Rating : 3 out of 5

Figure 5: Template 1: App review classification prompt for open-source models

Methodology : Prompts for Fine-Tuning Custom Model

```
Task Description:

Review user reviews for mobile applications based on their content, sentiment, and ratings. Utilize the definitions provided to classify each
review into the appropriate category.

Definitions for Classification:

Bug Reports:
Definition: Bug reports are user comments that identify issues with the app, such as crashes, incorrect behavior, or performance problems. These
reviews specifically highlight problems that affect the app's functionality and suggest a need for corrective action.

Feature Requests:
Definition: Feature requests are suggestions by users for new features or enhancements in future app updates. These can include requests for
features seen in other apps, additions to content, or ideas to modify existing features to enhance user interaction and satisfaction.

User Experience:
Definition: User experience reviews provide detailed narratives focusing on specific app features and their effectiveness in real scenarios.
They offer insights into the app's usability, functionality, and overall satisfaction, often serving as informal documentation of user needs
and app performance.
Differentiating Tip: Prioritize reviews that give detailed explanations of the app's features and their practical impact on the user.

Ratings:
Definition: Ratings are brief textual comments that reflect the app's numeric star rating, primarily indicating overall user satisfaction or
dissatisfaction. These reviews are succinct, focusing on expressing a general sentiment without detailed justification.
Differentiating Tip: Focus on reviews that lack detailed discussion of specific features or user experiences, and instead provide general
expressions of approval or disapproval.

Instructions to the Language Model:

Review Processing: Carefully read the provided app review and its star rating.
Give a brief explanation of the classification decision made for the review and Classify the review into one of the following
categories: "Bug", "Feature", "UserExperience", or "Rating".

Output Format: Provide the classification results in the following JSON format:
{{
    "Explanation": "<explanation>",
    "Class": "<predition>"
}}

Review and Star Rating to Classify:
User Review : "Absolutely handy for those pics you don't need everyone else to see."
User Rating : 3 out of 5
```


Figure 6: Template 2: App review classification prompt for open-source models

Methodology :   Fine-Tuning Open Source Models

Key Components:

- **Tool:** Hugging Face `SFTTrainer` Library
- **Hardware:** Consumer-grade GPU (24 GB VRAM)

Optimization Techniques:

- 4-bit quantization (QLoRA) [12]
- PEFT (Parameter-Efficient Fine-Tuning) [13]

[12] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. Preprint, arXiv:2305.14314.

[13] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter- efficient fine-tuning methods. https://github.com/huggingface/peft

Methodology : Evaluation Strategy

- Three experimental runs per experiment, averaged results
- Resubmission of invalid LLM responses until valid JSON obtained
- Metrics: Precision, Recall, F1-score (macro-averaged)
- Manual review of auto-annotated dataset
  - Sample size: 370 (Krejcie and Morgan Table)
  - Three annotators, Cohen's kappa for agreement
  - Majority label used, discussions for ties
- Accuracy evaluation of generated explanations (explain-then-annotate pattern)

Experiments and results:Evaluating Commercial Model Performance

Model Name	Bugs			Feature			Userexperience			Rating			Macro Avg		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Gemini Pro	1.00000	0.69333	0.81642	0.96270	0.66000	0.78264	0.67373	0.89333	0.76808	0.91623	0.50667	0.65246	0.81142	0.76333	0.75490
GPT 3.5 Turbo	0.83261	0.92667	0.87701	0.84969	0.82667	0.83788	0.87150	0.86000	0.86557	0.84871	0.78667	0.81624	0.85063	0.85000	0.84917
Base llama	0.60318	0.88000	0.71542	0.88189	0.28667	0.43142	0.67024	0.48667	0.56284	0.59229	0.74667	0.66046	0.66440	0.60000	0.57753
Base mistral	0.66769	0.73333	0.69882	0.63743	0.22000	0.32649	0.40545	0.80000	0.53798	0.43591	0.25333	0.31912	0.53662	0.50167	0.47060
llama + instruct finetune (10k)	0.84212	0.88667	0.86375	0.78199	0.86000	0.81910	0.85761	0.92000	0.88759	0.87924	0.68000	0.76620	0.84024	0.83667	0.83416
mistral + instruct finetune (10k)	0.85926	0.77333	0.81404	0.74127	0.92667	0.82294	0.77677	0.88000	0.82482	0.87438	0.62000	0.72356	0.81292	0.80000	0.79634
llama + instruct finetune (10k) + explanation	0.83144	0.88667	0.85794	0.74492	0.83333	0.78637	0.85523	0.89333	0.87385	0.85480	0.66667	0.74837	0.82410	0.82000	0.81786
mistral + instruct finetune (10k) + explanation	0.81092	0.85333	0.83119	0.72597	0.84667	0.78152	0.88876	0.90000	0.89410	0.89881	0.68667	0.77778	0.83112	0.82167	0.82115

Table 1: Model Performance Comparison Including Gemini Pro and GPT 3.5 Turbo

Experiments and results:Evaluating Commercial Model Performance

- Tested GPT-3.5 and Gemini Pro 1.0 in zero-shot setting
- F1 scores: GPT-3.5 (0.84917), Gemini Pro (0.75490)

Investigated impact of Temperature and `Top_p` parameters

- Lower values generally improved performance
- GPT-3.5 more responsive to parameter changes

Experiments and results:Evaluating Commercial Model Performance

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
	0	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.25	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
0	0.5	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.75	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67350	0.88000	0.76302	0.93021	0.53333	0.67792	0.81984	0.77333	0.76685
	1	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.25	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
0.5	0.5	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67350	0.88000	0.76302	0.93021	0.53333	0.67792	0.81984	0.77333	0.76685
	0.75	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67703	0.88000	0.76517	0.90857	0.52667	0.66631	0.81531	0.77167	0.76448
	1	0.70097	1.00000	0.82419	0.98095	0.68000	0.80317	0.64902	0.88000	0.74631	0.91098	0.47333	0.62043	0.81048	0.75833	0.74853
	0	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.25	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
1	0.5	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67514	0.88667	0.76657	0.94130	0.53333	0.68084	0.82302	0.77500	0.76846
	0.75	0.70097	1.00000	0.82419	0.98095	0.68000	0.80317	0.68795	0.88000	0.77209	0.91161	0.54667	0.68299	0.82037	0.77667	0.77061
	1	0.69301	0.99333	0.81642	0.96270	0.66000	0.78264	0.67373	0.89333	0.76808	0.91623	0.50667	0.65246	0.81142	0.76333	0.75490
	0	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.25	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
1.5	0.5	0.70423	1.00000	0.82645	0.97115	0.67333	0.79524	0.68019	0.89333	0.77231	0.94212	0.54000	0.68647	0.82442	0.77667	0.77012
	0.75	0.70097	1.00000	0.82419	0.98095	0.67333	0.79839	0.68331	0.92000	0.78406	0.97575	0.52667	0.68395	0.83524	0.78000	0.77265
	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
	0.25	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67692	0.88000	0.76522	0.93103	0.54000	0.68354	0.82090	0.77500	0.76880
2	0.5	0.70423	1.00000	0.82645	0.97143	0.68000	0.80000	0.67677	0.89333	0.77011	0.95238	0.53333	0.68376	0.82620	0.77667	0.77008
	0.75	0.69770	1.00000	0.82193	0.99020	0.67333	0.80159	0.70346	0.92667	0.79941	0.97536	0.55333	0.70412	0.84168	0.78833	0.78176
	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 6: Effects of Temperature and Top_p on Model Performance Metrics of Gemini Pro 1.0

Experiments and results:Evaluating Commercial Model Performance

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	0	0.85040	0.94667	0.89589	0.86274	0.86667	0.86415	0.88745	0.89333	0.89036	0.91599	0.80000	0.85360	0.87915	0.87667	0.87600
	0.25	0.85538	0.94667	0.89865	0.85136	0.87333	0.86208	0.85809	0.88667	0.87213	0.89605	0.74667	0.81454	0.86522	0.86333	0.86185
	0.5	0.85567	0.94667	0.89876	0.86768	0.87333	0.87042	0.89436	0.90000	0.89709	0.90165	0.79333	0.84396	0.87984	0.87833	0.87756
	0.75	0.84642	0.95333	0.89660	0.87797	0.86000	0.86865	0.88190	0.89333	0.88744	0.90175	0.79333	0.84386	0.87701	0.87500	0.87414
	1	0.86147	0.95333	0.90506	0.84527	0.87333	0.85906	0.90703	0.90667	0.90642	0.89978	0.77333	0.83148	0.87839	0.87667	0.87550
0.5	0	0.847470	0.960000	0.900120	0.871560	0.860000	0.865720	0.880300	0.880000	0.879980	0.893910	0.786670	0.836790	0.873310	0.871670	0.870650
	0.25	0.836330	0.953330	0.890960	0.853330	0.853330	0.853330	0.892650	0.886670	0.889630	0.915370	0.793330	0.849820	0.874420	0.871670	0.870940
	0.5	0.834380	0.940000	0.884030	0.809170	0.846670	0.827380	0.892760	0.886670	0.889620	0.903990	0.753330	0.821790	0.860070	0.856670	0.855700
	0.75	0.840330	0.946670	0.890320	0.860360	0.860000	0.860120	0.911430	0.860000	0.883970	0.858180	0.793330	0.822850	0.867580	0.865000	0.864310
	1	0.835560	0.946670	0.887540	0.842110	0.846670	0.844030	0.890560	0.866670	0.878370	0.872050	0.773330	0.819710	0.860070	0.858330	0.857410
1	0	0.849830	0.940000	0.892530	0.855540	0.866670	0.860820	0.889310	0.906670	0.897730	0.914730	0.786670	0.845880	0.877350	0.875000	0.874240
	0.25	0.852310	0.960000	0.902890	0.861140	0.866670	0.863820	0.905560	0.893330	0.899320	0.886360	0.780000	0.829790	0.876340	0.875000	0.873950
	0.5	0.842110	0.960000	0.897200	0.860680	0.860000	0.860170	0.871530	0.900000	0.885300	0.919600	0.760000	0.832030	0.873480	0.870000	0.868670
	0.75	0.836530	0.953330	0.891080	0.871480	0.853330	0.861880	0.889740	0.860000	0.874600	0.867540	0.793330	0.828360	0.866320	0.865000	0.863980
	1	0.832610	0.926670	0.877010	0.849690	0.826670	0.837880	0.871500	0.860000	0.865570	0.848710	0.786670	0.816240	0.850630	0.850000	0.849180
1.5	0	0.852130	0.960000	0.902840	0.871910	0.860000	0.865770	0.894190	0.900000	0.896950	0.901520	0.793330	0.843970	0.879940	0.878330	0.877380
	0.25	0.844630	0.940000	0.889660	0.832200	0.860000	0.845860	0.893310	0.893330	0.893260	0.914300	0.780000	0.841650	0.871110	0.868330	0.867600
	0.5	0.846250	0.953330	0.896540	0.860230	0.860000	0.860060	0.875100	0.886670	0.880710	0.899190	0.773330	0.831450	0.870190	0.868330	0.867190
	0.75	0.842880	0.926670	0.882640	0.842440	0.853330	0.847620	0.888890	0.853330	0.870750	0.863090	0.800000	0.830330	0.859330	0.858330	0.857830
	1	0.822700	0.920000	0.868220	0.804950	0.820000	0.812240	0.809280	0.733330	0.769050	0.720080	0.686670	0.702910	0.789250	0.790000	0.788100
2	0	0.845310	0.946670	0.893080	0.849890	0.866670	0.858140	0.904600	0.873330	0.888370	0.896070	0.800000	0.844870	0.873970	0.871670	0.871110
	0.25	0.845200	0.946670	0.893020	0.865850	0.860000	0.862900	0.889580	0.906670	0.897650	0.924070	0.800000	0.856920	0.881180	0.878330	0.877620
	0.5	0.846420	0.953330	0.896600	0.845320	0.866670	0.855350	0.875490	0.886670	0.880910	0.912120	0.760000	0.829110	0.869840	0.866670	0.865490
	0.75	0.844740	0.940000	0.889710	0.854430	0.860000	0.857080	0.886450	0.860000	0.871710	0.891920	0.806670	0.845560	0.869380	0.866670	0.866020
	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7: Effects of Temperature and Top_p on Model Performance Metrics of GPT 3.5

Experiments and results:Evaluating Quality of GPT-3.5 annotated dataset

Quality assessment of GPT-3.5 annotated dataset:

- 370 sample reviews (95% confidence, 5% margin of error)
- Inter-annotator agreement: $\kappa = 0.9135$ (almost perfect)
- Dataset accuracy: 0.8189.

Annotator	Bug	Feature	Rating	UserExp
Annotator 1	84	55	84	147
Annotator 2	82	69	89	130
Annotator 3	84	68	83	135

Table 5: Annotation Distribution by Annotator and Category

Annotator Pair	Kappa Score	Agreement Level
Annotator 1 vs 2	0.9146	Almost perfect
Annotator 1 vs 3	0.9180	Almost perfect
Annotator 2 vs 3	0.9079	Almost perfect
Average	0.9135	Almost perfect

Table 4: Pairwise Cohen's Kappa Scores and Agreement Levels

Experiments and results:Evaluating Open Source Models

Models tested: Llama 2, Mistral (Falcon excluded due to formatting issues)
Base model performance (F1 scores):

- Llama 2: 0.57753
- Mistral: 0.47060

Instruction fine-tuning:

- Used 10,000 GPT-3.5 annotated samples
- Two prompt templates tested
- Llama 2 performed best with Template 1
- Mistral excelled with "explain-then-annotate" Template 2

Experiments and results:Evaluating Open Source Models

Training optimization:

- Performance improved with larger dataset sizes
- Fewer samples with multiple epochs ≈ More samples with fewer epochs

Experiments and results:Evaluating Open Source Models

Training Sample Size	Model Name	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
(a) Without label explanations in the prompt																
10000 1 epoch	llama	0.84212	0.88667	0.86375	0.78199	0.86000	0.81910	0.85761	0.92000	0.88759	0.87924	0.68000	0.76620	0.84024	0.83667	0.83416
	mistral	0.85926	0.77333	0.81404	0.74127	0.92667	0.82294	0.77677	0.88000	0.82482	0.87430	0.62000	0.72356	0.81290	0.80000	0.79634
8000 1 epoch	llama	0.83716	0.92667	0.87953	0.78139	0.80000	0.78973	0.91032	0.86667	0.88768	0.79660	0.72667	0.75813	0.83136	0.83000	0.82877
	mistral	0.82840	0.89333	0.85899	0.71691	0.85333	0.77857	0.90198	0.85333	0.87683	0.80189	0.62667	0.70287	0.81230	0.80667	0.80432
6000 1 epoch	llama	0.91069	0.72667	0.80738	0.85391	0.84667	0.84966	0.79138	0.92667	0.85303	0.74222	0.76667	0.75342	0.82455	0.81667	0.81587
	mistral	0.89478	0.79333	0.84099	0.78171	0.88000	0.82789	0.78754	0.86000	0.82176	0.76807	0.68667	0.72446	0.80802	0.80500	0.80377
4000 1 epoch	llama	0.68359	0.96000	0.79824	0.89821	0.52000	0.65758	0.82189	0.79333	0.80723	0.75780	0.79333	0.77510	0.79037	0.76667	0.75953
	mistral	0.87606	0.84000	0.85681	0.85662	0.82000	0.83714	0.75278	0.86000	0.80189	0.76785	0.71333	0.73811	0.81333	0.80833	0.80849
2000 1 epoch	llama	0.77242	0.92667	0.84239	0.90623	0.70667	0.79272	0.73109	0.92000	0.81451	0.87722	0.66667	0.75652	0.82174	0.80500	0.80153
	mistral	0.82115	0.73333	0.77467	0.90922	0.73333	0.81150	0.59876	0.93333	0.72946	0.75237	0.56000	0.64179	0.77038	0.74000	0.73935
1500 1 epoch	llama	0.86375	0.84000	0.85106	0.62651	0.86000	0.72475	0.82639	0.86000	0.84223	0.83390	0.50667	0.62787	0.78764	0.76667	0.76148
	mistral	0.77689	0.88000	0.82518	0.68465	0.78000	0.72879	0.83887	0.68667	0.75468	0.70663	0.64000	0.67134	0.75176	0.74667	0.74500
1000 1 epoch	llama	0.84028	0.82667	0.83196	0.69499	0.88667	0.77829	0.74260	0.77333	0.75690	0.69502	0.48000	0.56660	0.74322	0.74167	0.73344
	mistral	0.73160	0.85333	0.78740	0.70044	0.75333	0.72428	0.77408	0.44667	0.56430	0.57742	0.67333	0.62114	0.69588	0.68167	0.67428
500 1 epoch	llama	0.75900	0.90000	0.82346	0.62743	0.82000	0.71078	0.63354	0.85333	0.72716	0.59259	0.09333	0.15990	0.65314	0.66667	0.60533
	mistral	0.78667	0.82667	0.80590	0.62141	0.79333	0.69668	0.66082	0.74000	0.69806	0.65397	0.36000	0.46352	0.68071	0.68000	0.66604
(b) With label explanations in the prompt																
10000 1 epoch + Explanation	llama	0.83144	0.88667	0.85794	0.74492	0.83333	0.78637	0.86523	0.89333	0.87882	0.85480	0.66667	0.74837	0.82410	0.82000	0.81788
	mistral	0.81092	0.85333	0.83119	0.72597	0.84667	0.78152	0.88876	0.90000	0.89410	0.89881	0.68667	0.77778	0.83112	0.82167	0.82115
8000 1 epoch + Explanation	llama	0.77874	0.95333	0.85667	0.82365	0.75333	0.78381	0.94259	0.84667	0.89139	0.83339	0.79333	0.81113	0.84459	0.83667	0.83575
	mistral	0.83647	0.81333	0.82437	0.74000	0.81333	0.77481	0.81173	0.91333	0.85909	0.82413	0.66000	0.73277	0.80309	0.80000	0.79776
6000 1 epoch + Explanation	llama	0.84879	0.82000	0.83378	0.85209	0.74000	0.79109	0.71503	0.94667	0.81394	0.79033	0.66000	0.71697	0.80156	0.79167	0.78894
	mistral	0.83219	0.66000	0.73546	0.80097	0.78667	0.79232	0.65468	0.90667	0.76009	0.70443	0.58667	0.63985	0.74807	0.73500	0.73193
4000 1 epoch + Explanation	llama	0.68596	0.96000	0.80006	0.82963	0.67333	0.74267	0.89670	0.79333	0.84099	0.80734	0.72667	0.76355	0.80490	0.78833	0.78682
	mistral	0.74446	0.86667	0.79992	0.85739	0.68000	0.75803	0.80125	0.91333	0.85350	0.82662	0.74000	0.77950	0.80743	0.80000	0.79774
2000 1 epoch + Explanation	llama	0.72317	0.90000	0.80162	0.84313	0.61333	0.70870	0.78207	0.94667	0.85611	0.79459	0.64667	0.71189	0.78574	0.77667	0.76958
	mistral	0.70319	0.90000	0.78930	0.91499	0.57333	0.70491	0.73172	0.90667	0.80784	0.78314	0.66000	0.71526	0.78326	0.76000	0.75433
1500 1 epoch + Explanation	llama	0.75800	0.91333	0.82800	0.68278	0.78667	0.73096	0.88903	0.88000	0.88345	0.88668	0.57333	0.69635	0.80412	0.78833	0.78469
	mistral	0.80024	0.84667	0.82147	0.66083	0.79333	0.72095	0.80488	0.93333	0.86423	0.82316	0.48000	0.60566	0.77228	0.76333	0.75308
1000 1 epoch + Explanation	llama	0.82163	0.88667	0.85256	0.77295	0.85333	0.80976	0.75984	0.84000	0.79782	0.80175	0.56667	0.66212	0.78905	0.78667	0.78056
	mistral	0.80321	0.89333	0.84559	0.72971	0.78667	0.75642	0.72553	0.88000	0.79473	0.85381	0.50667	0.63568	0.77806	0.76667	0.75811
0500 1 epoch + Explanation	llama	0.77581	0.87333	0.82126	0.76586	0.70000	0.73042	0.65362	0.90000	0.75704	0.76717	0.44667	0.56236	0.74062	0.73000	0.71777
	mistral	0.85296	0.77333	0.81091	0.81427	0.81333	0.81309	0.63861	0.95333	0.76463	0.79023	0.47333	0.59175	0.77402	0.75333	0.74509

Table 2: Model Performance Comparison With and Without Label Explanations in the Prompt

Experiments and results:Evaluating Open Source Models

Training Sample Size	Model Name	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
(a) Without label explanations in the prompt																
2000 1 epoch	llama	0.77242	0.92667	0.84239	0.90623	0.70667	0.79272	0.73109	0.92000	0.81451	0.87722	0.66667	0.75652	0.82174	0.80500	0.80153
	mistral	0.82115	0.73333	0.77467	0.90922	0.73333	0.81150	0.59876	0.93333	0.72946	0.75237	0.56000	0.64179	0.77038	0.74000	0.73935
2000 2 epoch	llama	0.92097	0.78000	0.84454	0.83281	0.79333	0.81234	0.87657	0.83333	0.85400	0.68463	0.85333	0.75942	0.82875	0.81500	0.81758
	mistral	0.86996	0.84667	0.85802	0.89855	0.82667	0.86111	0.88151	0.74667	0.80787	0.71271	0.89333	0.79230	0.84068	0.82833	0.82982
2000 3 epoch	llama	0.86843	0.82667	0.84663	0.81624	0.82667	0.82113	0.72137	0.90667	0.80306	0.82712	0.64000	0.71965	0.80829	0.80000	0.79762
	mistral	0.91007	0.80667	0.85480	0.86773	0.82667	0.84644	0.74995	0.93333	0.83131	0.79634	0.72667	0.75965	0.83102	0.82333	0.82305
2000 4 epoch	llama	0.87827	0.86000	0.86886	0.82239	0.82667	0.82384	0.88889	0.86667	0.87581	0.81278	0.84000	0.82595	0.85058	0.84833	0.84862
	mistral	0.88675	0.78000	0.82975	0.75054	0.89333	0.81508	0.90089	0.78000	0.83539	0.80544	0.85333	0.82809	0.83591	0.82667	0.82708
(b) With label explanations in the prompt																
2000 1 epoch + Explanation	llama	0.72317	0.90000	0.80162	0.84313	0.61333	0.70870	0.78207	0.94667	0.85611	0.79459	0.64667	0.71189	0.78574	0.77667	0.76958
	mistral	0.70319	0.90000	0.78930	0.91499	0.57333	0.70491	0.73172	0.90667	0.80784	0.78314	0.66000	0.71526	0.78326	0.76000	0.75433
2000 2 epoch + Explanation	llama	0.84654	0.80667	0.82598	0.81867	0.80667	0.81206	0.93663	0.80000	0.86182	0.72556	0.87333	0.79201	0.83185	0.82167	0.82297
	mistral	0.85470	0.82000	0.83693	0.79603	0.82667	0.81070	0.89807	0.82000	0.85689	0.75678	0.82000	0.78572	0.82640	0.82167	0.82256
2000 3 epoch + Explanation	llama	0.88934	0.77333	0.82472	0.84636	0.88000	0.86280	0.68674	0.96667	0.80233	0.89832	0.60667	0.72296	0.83019	0.80667	0.80320
	mistral	0.90242	0.68667	0.77964	0.86148	0.84000	0.84956	0.61443	0.98667	0.75723	0.87756	0.57333	0.69310	0.81397	0.77167	0.76988
2000 4 epoch + Explanation	llama	0.87177	0.86000	0.86571	0.84096	0.80667	0.82340	0.80829	0.92667	0.86338	0.83194	0.75333	0.79054	0.83824	0.83667	0.83576
	mistral	0.88175	0.84000	0.85979	0.82985	0.86667	0.84737	0.87276	0.91333	0.89235	0.81136	0.77333	0.79175	0.84893	0.84833	0.84782

Table 3: Model Performance Comparison With and Without Label Explanations in the Prompt

Experiments and results:Evaluating Open Source Models

Mistral explanation quality (168 correct classifications):

- Grade A: 76.79% (129)
- Grade B: 16.67% (28)
- Grade C: 5.95% (10)
- Grade D: 0.60% (1)
- 93.45% satisfactory results (A or B)

`Temperature` and `top_p` effects similar to commercial LLMs

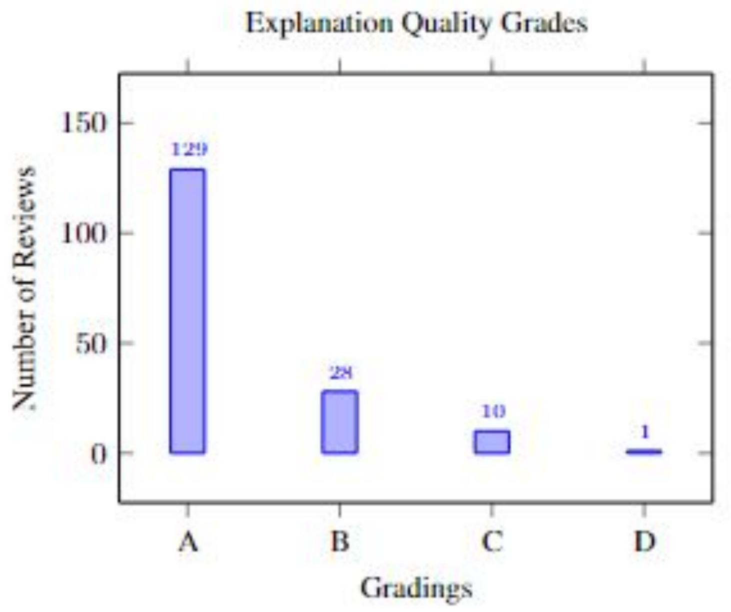


Figure 3: Distribution of grades for explanations generated by the fine-tuned Mistral 7B model on our 200-review benchmark dataset, evaluated by human experts. Grades range from A (highest quality) to D (lowest quality).

Experiments and results:Evaluating Open Source Models

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	0	0.85185	0.92000	0.88462	0.82323	0.90000	0.85989	0.89040	0.92000	0.90494	0.92433	0.73333	0.81771	0.87245	0.86833	0.86679
	0.25	0.84669	0.92000	0.88181	0.82211	0.89333	0.85623	0.87905	0.92000	0.89904	0.92304	0.72000	0.80889	0.86772	0.86333	0.86149
	0.5	0.85092	0.91333	0.88101	0.80861	0.90000	0.85180	0.88462	0.92000	0.90196	0.92240	0.71333	0.80448	0.86664	0.86167	0.85981
	0.75	0.84669	0.92000	0.88181	0.81706	0.89333	0.85348	0.87905	0.92000	0.89904	0.92240	0.71333	0.80448	0.86630	0.86167	0.85970
	1	0.84153	0.92000	0.87900	0.81261	0.89333	0.85088	0.87264	0.91333	0.89251	0.92091	0.70000	0.79504	0.86192	0.85667	0.85436
0.5	0	0.84669	0.92000	0.88181	0.81706	0.89333	0.85348	0.87905	0.92000	0.89904	0.92240	0.71333	0.80448	0.86630	0.86167	0.85970
	0.25	0.84669	0.92000	0.88181	0.82716	0.89333	0.85897	0.87905	0.92000	0.89904	0.92372	0.72667	0.81340	0.86915	0.86500	0.86331
	0.5	0.84052	0.91333	0.87540	0.80606	0.88667	0.84444	0.88462	0.92000	0.90196	0.92240	0.71333	0.80448	0.86340	0.85833	0.85657
	0.75	0.85120	0.91333	0.88105	0.78832	0.89333	0.83754	0.87905	0.92000	0.89904	0.91955	0.68667	0.78601	0.85953	0.85333	0.85091
	1	0.85636	0.91333	0.88386	0.80754	0.89333	0.84821	0.89137	0.92667	0.90862	0.91538	0.72000	0.80599	0.86766	0.86333	0.86167
1	0	0.85092	0.91333	0.88101	0.81331	0.90000	0.85445	0.87349	0.92000	0.89612	0.92173	0.70667	0.79997	0.86486	0.86000	0.85789
	0.25	0.84669	0.92000	0.88181	0.81706	0.89333	0.85348	0.88483	0.92000	0.90202	0.92304	0.72000	0.80889	0.86791	0.86333	0.86155
	0.5	0.84688	0.92000	0.88186	0.80613	0.88667	0.84444	0.86792	0.92000	0.89320	0.92034	0.69333	0.79084	0.86032	0.85500	0.85259
	0.75	0.84285	0.89333	0.86731	0.77307	0.88000	0.82277	0.89062	0.92000	0.90500	0.90526	0.69333	0.78470	0.85295	0.84667	0.84494
	1	0.84212	0.88667	0.86375	0.78199	0.86000	0.81910	0.85761	0.92000	0.88759	0.87924	0.68000	0.76620	0.84024	0.83667	0.83416
1.5	0	0.85092	0.91333	0.88101	0.81331	0.90000	0.85445	0.89062	0.92000	0.90500	0.92368	0.72667	0.81330	0.86963	0.86500	0.86344
	0.25	0.85185	0.92000	0.88462	0.81818	0.90000	0.85714	0.88462	0.92000	0.90196	0.92308	0.72000	0.80899	0.86943	0.86500	0.86318
	0.5	0.84917	0.93333	0.88911	0.82611	0.88667	0.85530	0.86813	0.92000	0.89326	0.90413	0.69333	0.78471	0.86189	0.85833	0.85560
	0.75	0.83809	0.89333	0.86460	0.81077	0.88000	0.84372	0.84616	0.90667	0.87467	0.88761	0.68667	0.77289	0.84566	0.84167	0.83897
	1	0.83015	0.88000	0.85430	0.73052	0.82667	0.77513	0.84925	0.86000	0.85435	0.80136	0.63333	0.70526	0.80282	0.80000	0.79726
2	0	0.85185	0.92000	0.88462	0.82323	0.90000	0.85989	0.89040	0.92000	0.90494	0.92436	0.73333	0.81781	0.87246	0.86833	0.86681
	0.25	0.85185	0.92000	0.88462	0.82323	0.90000	0.85989	0.87927	0.92000	0.89910	0.92304	0.72000	0.80889	0.86935	0.86500	0.86312
	0.5	0.85383	0.88667	0.86955	0.79173	0.88667	0.83647	0.87207	0.90667	0.88864	0.91777	0.73333	0.81492	0.85885	0.85333	0.85239
	0.75	0.80891	0.87333	0.83946	0.75186	0.80667	0.77824	0.83998	0.90667	0.87190	0.84345	0.64667	0.73204	0.81105	0.80833	0.80541
	1	0.83641	0.82000	0.82807	0.68624	0.85333	0.76033	0.77526	0.87333	0.82098	0.77627	0.50000	0.60611	0.76854	0.76167	0.75387

Table 8: Effects of Temperature and Top_p on Model Performance Metrics of LLMA 2 instruct fine-tuned using Prompt Template A.2

Experiments and results:Evaluating Open Source Models

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	0	0.83019	0.88000	0.85437	0.81151	0.86000	0.83500	0.88132	0.94000	0.90970	0.92616	0.75333	0.83066	0.86229	0.85833	0.85743
	0.25	0.81994	0.88000	0.84889	0.79881	0.84667	0.82200	0.88607	0.93333	0.90907	0.91809	0.74667	0.82352	0.85573	0.85167	0.85087
	0.5	0.82506	0.88000	0.85163	0.80510	0.85333	0.82847	0.87512	0.93333	0.90322	0.91761	0.74000	0.81910	0.85572	0.85167	0.85060
	0.75	0.82506	0.88000	0.85163	0.80510	0.85333	0.82847	0.88210	0.94667	0.91318	0.93368	0.74667	0.82950	0.86149	0.85667	0.85569
	1	0.83039	0.88000	0.85442	0.81938	0.84667	0.83278	0.89431	0.96000	0.92587	0.93612	0.78000	0.85093	0.87005	0.86667	0.86600
0.5	0	0.82506	0.88000	0.85163	0.81011	0.85333	0.83114	0.88132	0.94000	0.90970	0.92622	0.75333	0.83085	0.86068	0.85667	0.85583
	0.25	0.82506	0.88000	0.85163	0.80510	0.85333	0.82847	0.88679	0.94000	0.91262	0.92622	0.75333	0.83085	0.86079	0.85667	0.85589
	0.5	0.83019	0.88000	0.85437	0.81132	0.86000	0.83495	0.87421	0.92667	0.89968	0.91057	0.74667	0.82051	0.85657	0.85333	0.85238
	0.75	0.82611	0.88667	0.85530	0.81406	0.84667	0.82999	0.86627	0.94667	0.90457	0.92484	0.73333	0.81767	0.85782	0.85333	0.85188
	1	0.81366	0.90000	0.85455	0.78323	0.84000	0.81012	0.88157	0.93333	0.90650	0.93065	0.70667	0.80297	0.85228	0.84500	0.84354
1	0	0.83551	0.88000	0.85716	0.80130	0.86000	0.82960	0.88677	0.94000	0.91255	0.91809	0.74667	0.82352	0.86042	0.85667	0.85571
	0.25	0.83019	0.88000	0.85437	0.80130	0.86000	0.82960	0.88679	0.94000	0.91262	0.92561	0.74667	0.82654	0.86097	0.85667	0.85578
	0.5	0.83039	0.88000	0.85442	0.83044	0.84667	0.83832	0.87346	0.96667	0.91763	0.95155	0.77333	0.85277	0.87146	0.86667	0.86579
	0.75	0.82218	0.89333	0.85622	0.78179	0.85333	0.81558	0.85834	0.92667	0.89114	0.91880	0.68000	0.78128	0.84528	0.83833	0.83606
	1	0.83144	0.88667	0.85794	0.74492	0.83333	0.78637	0.86523	0.89333	0.87882	0.85480	0.66667	0.74837	0.82410	0.82000	0.81788
1.5	0	0.83039	0.88000	0.85442	0.81155	0.86000	0.83492	0.88607	0.93333	0.90907	0.91960	0.76000	0.83203	0.86190	0.85833	0.85761
	0.25	0.83019	0.88000	0.85437	0.80631	0.86000	0.83228	0.87584	0.94000	0.90677	0.92497	0.74000	0.82213	0.85933	0.85500	0.85389
	0.5	0.82735	0.89333	0.85903	0.79226	0.84000	0.81524	0.84426	0.94000	0.88955	0.93878	0.70000	0.80178	0.85066	0.84333	0.84140
	0.75	0.84483	0.90667	0.87459	0.78302	0.88667	0.83152	0.86659	0.90667	0.88594	0.92022	0.68667	0.78606	0.85367	0.84667	0.84453
	1	0.84492	0.80000	0.82170	0.68729	0.89333	0.77676	0.81761	0.78667	0.80034	0.82385	0.64667	0.72198	0.79342	0.78167	0.78020
2	0	0.81994	0.88000	0.84889	0.79874	0.84667	0.82201	0.88607	0.93333	0.90907	0.91809	0.74667	0.82352	0.85571	0.85167	0.85087
	0.25	0.81994	0.88000	0.84889	0.80890	0.84667	0.82734	0.88459	0.92000	0.90189	0.90510	0.76000	0.82615	0.85463	0.85167	0.85107
	0.5	0.82043	0.90667	0.86110	0.78093	0.84667	0.81141	0.87369	0.92000	0.89618	0.92123	0.69333	0.79108	0.84907	0.84167	0.83994
	0.75	0.83499	0.90000	0.86549	0.72057	0.84000	0.77446	0.84427	0.86667	0.85498	0.82668	0.60000	0.69406	0.80663	0.80167	0.79725
	1	0.80849	0.84000	0.82358	0.63904	0.84667	0.72818	0.85617	0.74667	0.79671	0.67698	0.51333	0.58343	0.74517	0.73667	0.73298

Table 9: Effects of Temperature and Top_p on Model Performance Metrics of LLMA 2 instruct fine-tuned using *explain-then-annotate* Prompt Template A.3

Experiments and results:Evaluating Open Source Models

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	0	0.83974	0.87333	0.85621	0.84382	0.90000	0.87099	0.82190	0.95333	0.88271	0.94545	0.69333	0.79996	0.86273	0.85500	0.85247
	0.25	0.83974	0.87333	0.85621	0.84382	0.90000	0.87099	0.82190	0.95333	0.88271	0.94545	0.69333	0.79996	0.86273	0.85500	0.85247
	0.5	0.83974	0.87333	0.85621	0.84906	0.90000	0.87379	0.81742	0.95333	0.88011	0.94542	0.69333	0.79986	0.86291	0.85500	0.85249
	0.75	0.83556	0.84667	0.84106	0.82828	0.90000	0.86264	0.81723	0.95333	0.87999	0.94539	0.69333	0.79975	0.85661	0.84833	0.84586
	1	0.83571	0.88000	0.85721	0.84277	0.89333	0.86731	0.82982	0.94000	0.88137	0.93812	0.70667	0.80608	0.86160	0.85500	0.85299
0.5	0	0.84013	0.87333	0.85625	0.83333	0.90000	0.86538	0.82128	0.94667	0.87934	0.94494	0.68667	0.79533	0.85992	0.85167	0.84908
	0.25	0.83450	0.87333	0.85341	0.84906	0.90000	0.87379	0.80577	0.94000	0.86771	0.94494	0.68667	0.79533	0.85857	0.85000	0.84756
	0.5	0.83442	0.87333	0.85342	0.84926	0.90000	0.87384	0.82574	0.94667	0.88202	0.94642	0.70667	0.80913	0.86396	0.85667	0.85460
	0.75	0.83002	0.84667	0.83821	0.84986	0.90000	0.87401	0.80119	0.94000	0.86505	0.93782	0.70000	0.80143	0.85472	0.84667	0.84467
	1	0.83465	0.80667	0.82029	0.79048	0.89333	0.83797	0.81572	0.94000	0.87336	0.94722	0.70667	0.80896	0.84702	0.83667	0.83514
1	0	0.83882	0.86667	0.85245	0.83352	0.90000	0.86544	0.82083	0.94667	0.87925	0.94542	0.69333	0.79986	0.85965	0.85167	0.84925
	0.25	0.83442	0.87333	0.85342	0.83857	0.90000	0.86819	0.82083	0.94667	0.87925	0.94494	0.68667	0.79533	0.85969	0.85167	0.84905
	0.5	0.86867	0.87333	0.87064	0.84475	0.90667	0.87460	0.84232	0.96000	0.89718	0.95747	0.74667	0.83894	0.87830	0.87167	0.87034
	0.75	0.87004	0.84667	0.85817	0.80400	0.87333	0.83717	0.82052	0.94000	0.87593	0.92457	0.73333	0.81652	0.85478	0.84833	0.84695
	1	0.85926	0.77333	0.81404	0.74127	0.92667	0.82294	0.77677	0.88000	0.82482	0.87430	0.62000	0.72356	0.81290	0.80000	0.79634
1.5	0	0.84515	0.87333	0.85899	0.84382	0.90000	0.87099	0.82291	0.96000	0.88617	0.94545	0.69333	0.79996	0.86433	0.85667	0.85403
	0.25	0.84414	0.86667	0.85524	0.84382	0.90000	0.87099	0.82184	0.95333	0.88272	0.94642	0.70667	0.80913	0.86405	0.85667	0.85452
	0.5	0.87170	0.86000	0.86555	0.82751	0.89333	0.85901	0.80593	0.94000	0.86776	0.92173	0.70667	0.79997	0.85671	0.85000	0.84807
	0.75	0.83668	0.78667	0.81084	0.75000	0.90000	0.81818	0.77073	0.89333	0.82735	0.89562	0.62667	0.73709	0.81326	0.80167	0.79837
	1	0.83244	0.78667	0.80863	0.69248	0.85333	0.76282	0.72974	0.82667	0.77492	0.80781	0.54667	0.64910	0.76562	0.75333	0.74887
2	0	0.83761	0.86000	0.84864	0.83857	0.90000	0.86819	0.82668	0.95333	0.88549	0.94642	0.70667	0.80913	0.86232	0.85500	0.85286
	0.25	0.83874	0.86667	0.85246	0.83857	0.90000	0.86819	0.82658	0.95333	0.88543	0.94595	0.70000	0.80460	0.86246	0.85500	0.85267
	0.5	0.83683	0.82000	0.82827	0.76624	0.87333	0.81625	0.85445	0.94000	0.89504	0.89804	0.70000	0.78667	0.83889	0.83333	0.83155
	0.75	0.79670	0.80000	0.79784	0.68130	0.82667	0.74683	0.72907	0.84000	0.78036	0.78544	0.49333	0.60505	0.74813	0.74000	0.73252
	1	0.75170	0.66667	0.70576	0.66257	0.80000	0.72366	0.69671	0.80667	0.74260	0.73931	0.53333	0.61870	0.71257	0.70167	0.69768

Table 10: Effects of `Temperature` and `Top_p` on Model Performance Metrics of `Mistral` instruct fine-tuned using Prompt Template A.2

Experiments and results:Evaluating Open Source Models

Temperature	Top_p	Bugs			Feature			Userexperience			Rating			Macro Avg		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	0	0.81836	0.90000	0.85719	0.75909	0.84000	0.79748	0.89370	0.94667	0.91920	0.94539	0.69333	0.79975	0.85413	0.84500	0.84341
	0.25	0.81212	0.89333	0.85079	0.75331	0.83333	0.79124	0.89248	0.94000	0.91560	0.94592	0.70000	0.80449	0.85096	0.84167	0.84053
	0.5	0.81543	0.91333	0.86157	0.78285	0.84000	0.81036	0.90007	0.96000	0.92905	0.94595	0.70000	0.80460	0.86107	0.85333	0.85139
	0.75	0.80267	0.89333	0.84529	0.77803	0.84000	0.80764	0.89318	0.94667	0.91909	0.93741	0.70000	0.80146	0.85282	0.84500	0.84337
	1	0.83256	0.89333	0.86176	0.77706	0.86000	0.81641	0.87733	0.95333	0.91374	0.93644	0.68667	0.79230	0.85584	0.84833	0.84605
0.5	0	0.81825	0.90000	0.85713	0.76364	0.84000	0.80000	0.89378	0.95333	0.92258	0.94545	0.69333	0.79996	0.85528	0.84667	0.84492
	0.25	0.81261	0.89333	0.85073	0.76964	0.84667	0.80628	0.89408	0.95333	0.92263	0.94545	0.69333	0.79996	0.85544	0.84667	0.84490
	0.5	0.80915	0.87333	0.83983	0.76836	0.84000	0.80248	0.88770	0.94667	0.91616	0.94734	0.72000	0.81808	0.85314	0.84500	0.84414
	0.75	0.81002	0.88000	0.84345	0.75611	0.84667	0.79879	0.89553	0.91333	0.90428	0.91502	0.70667	0.79709	0.84417	0.83667	0.83590
	1	0.80681	0.86000	0.83244	0.74541	0.84000	0.78978	0.85984	0.89333	0.87611	0.91314	0.70000	0.79237	0.83130	0.82333	0.82268
1	0	0.81846	0.90000	0.85711	0.76413	0.84000	0.80014	0.89308	0.94667	0.91909	0.94592	0.70000	0.80449	0.85540	0.84667	0.84521
	0.25	0.82360	0.90000	0.86000	0.76690	0.83333	0.79863	0.89378	0.95333	0.92258	0.94689	0.71333	0.81365	0.85779	0.85000	0.84872
	0.5	0.80511	0.88000	0.84083	0.77285	0.86000	0.81399	0.89868	0.94000	0.91863	0.94688	0.70667	0.80920	0.85588	0.84667	0.84566
	0.75	0.82906	0.87333	0.85055	0.74228	0.88000	0.80482	0.88401	0.90667	0.89495	0.95508	0.70000	0.80500	0.85261	0.84000	0.83883
	1	0.81092	0.85333	0.83119	0.72597	0.84667	0.78152	0.88876	0.90000	0.89410	0.89881	0.68667	0.77778	0.83112	0.82167	0.82115
1.5	0	0.80463	0.88000	0.84052	0.74886	0.83333	0.78879	0.87037	0.94000	0.90385	0.94392	0.67333	0.78596	0.84194	0.83167	0.82978
	0.25	0.81356	0.87333	0.84228	0.75909	0.84000	0.79748	0.91026	0.94667	0.92810	0.94036	0.73333	0.82395	0.85582	0.84833	0.84795
	0.5	0.82069	0.88000	0.84910	0.76504	0.86000	0.80911	0.87723	0.90000	0.88799	0.93053	0.72000	0.81074	0.84837	0.84000	0.83924
	0.75	0.83837	0.89333	0.86472	0.71015	0.86667	0.78061	0.87487	0.88667	0.88065	0.89568	0.62667	0.73671	0.82977	0.81833	0.81567
	1	0.80187	0.82000	0.80915	0.67014	0.86667	0.75571	0.89350	0.84667	0.86878	0.83939	0.61333	0.70698	0.80123	0.78667	0.78515
2	0	0.81731	0.89333	0.85352	0.76380	0.84000	0.80005	0.89387	0.95333	0.92257	0.94595	0.70000	0.80460	0.85523	0.84667	0.84519
	0.25	0.79411	0.90000	0.84356	0.79331	0.84000	0.81564	0.88700	0.94000	0.91268	0.94666	0.70667	0.80921	0.85527	0.84667	0.84527
	0.5	0.80239	0.81333	0.80770	0.72981	0.86000	0.78930	0.84530	0.91333	0.87795	0.92649	0.67333	0.77947	0.82600	0.81500	0.81361
	0.75	0.83902	0.76667	0.80107	0.64770	0.86667	0.74092	0.82554	0.86667	0.84498	0.82208	0.56667	0.67053	0.78358	0.76667	0.76438
	1	0.84335	0.75333	0.79497	0.58675	0.84000	0.69070	0.76639	0.74000	0.75170	0.72704	0.51333	0.60137	0.73088	0.71167	0.70969

Table 11: Effects of `Temperature` and `Top_p` on Model Performance Metrics of `Mistral` instruct fine-tuned using *explain-then-annotate* Prompt Template A.3

Conclusion

Explored commercial and open-source LLMs for app review classification
Key findings:

- Commercial LLMs effective in zero-shot settings
- Temperature and top_p parameters impact performance
- Fine-tuned open-source models show substantial gains
- Open-source models offer cost-effective alternative

Experiments conducted on:

- Training data size
- Number of epochs
- Temperature and top_p effects
- Quality of generated explanations

Resources published for further research
Future work: Investigate generalizability across multiple domains

Summary

Summary

1. Aspect-Based Sentiment Analysis (ABSA)
   - Productivity domain: F1 score 0.62 (+87.88%)
   - Gaming domain: F1 score 0.42 (+31.25%)
   - Social Networking: F1 score 0.62 (+93.75%)
   - Sentiment accuracy: 0.80, 0.70, 0.86 respectively
2. Embedding & Augmentation Analysis
   - Word2Vec outperformed alternatives (avg F1: 0.56)
   - RTT(DE) improved F1 scores by 2%
   - Optimal parameters identified:
   - - Batch size: 25/35 - Epochs: 75
   - Learning rate: 0.001
3. Large Language Model Applications
   - GPT-3.5 zero-shot: F1 score 0.84917
   - Autonomous annotation: 81.89% accuracy
   - Fine-tuned LLAMA 2: F1 score 0.83416
   - Fine-tuned Mistral: F1 score 0.82115
   - Cohen's Kappa: 0.9135 (inter-annotator agreement)

Published Work:
- ICTER 2022: ABSA findings
- Under review COLING 2025: LLM implementation

References

References

[1] “Number of smartphone users worldwide,” https://www.statista.com/statistics/ 330695/number-of-smartphone-users-worldwide/.
[2] L. V. G. Carreno and K. Winbladh, “Analysis of user comments: an approach for software requirements evolution,” in 2013 35th international conference on software engineering (ICSE). IEEE, 2013, pp. 582–591.
[3] E. Guzman and W. Maalej, “How do users like this feature? a fine grained sen timent analysis of app reviews,” in 2014 IEEE 22nd international requirements engineering conference (RE). Ieee, 2014, pp. 153–162.
[4] D. Pagano and B. Bruegge, “User involvement in software evolution practice: a case study,” in 2013 35th International Conference on Software Engineering (ICSE). IEEE, 2013, pp. 953–962.
[5] M. Harman, Y. Jia, and Y. Zhang, “App store mining and analysis: Msr for app stores,” in 2012 9th IEEE working conference on mining software repositories (MSR). IEEE, 2012, pp. 108–111.
[6] N.Chen, J. Lin, S. C. Hoi, X. Xiao, and B.Zhang, “Ar-miner: mining informative reviews for developers from mobile app marketplace,” in Proceedings of the 36th international conference on software engineering, 2014, pp. 767–778.
[7] F. Palomba, M. Linares-Vásquez, G. Bavota, R. Oliveto, M. Di Penta, D. Poshy vanyk, and A. De Lucia, “Crowdsourcing user reviews to support the evolution of mobile apps,” Journal of Systems and Software, vol. 137, pp. 143–162, 2018.
[8] E. Guzman, M. El-Haliby, and B. Bruegge, “Ensemble methods for app review classification: An approach for software evolution (n),” in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2015, pp. 771–776.
[9] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, “How can i improve my app? classifying user reviews for software maintenance and evolution,” in 2015 IEEE international conference on software maintenance and evolution (ICSME). IEEE, 2015, pp. 281–290.
[10] E. Guzman, M. Ibrahim, and M. Glinz, “A little bird told me: Mining tweets for requirements and software evolution,” 09 2017.
[11] N. Alturaief, H. Aljamaan, and M. Baslyman, “Aware: Aspect-based sentiment analysis dataset of apps reviews for requirements elicitation,” in 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW). IEEE, 2021, pp. 211–218.
[12] D.PaganoandW.Maalej,“User Feedback In The App Store: An Empirical Study,” in 2013 21st IEEE international requirements engineering conference (RE). IEEE, 2013, pp. 125–134.
[13] H. Li, L. Zhang, L. Zhang, and J. Shen, “A user satisfaction analysis approach for software evolution,” vol. 2, pp. 1093–1097, 2010.
[14] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, “Toward data-driven require ments engineering,” IEEE software, vol. 33, no. 1, pp. 48–54, 2015.
[15] M.V.Phong, T.T.Nguyen, H.V.Pham, and T.T.Nguyen, “Mining User Opinions in mobile app reviews: A keyword-based approach (t),” pp. 749–759, 2015.
[16] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, “Why people hate your app: Making sense of user feedback in a mobile app store,” pp. 1276–1284, 2013.
[17] R. T. Anchiêta and R. S. Moura, “Exploring unsupervised learning towards extractive summarization of user reviews,” in Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web, 2017, pp. 217–220.
[18] M. Gomez, R. Rouvoy, M. Monperrus, and L. Seinturier, “A recommender sys tem of buggy app checkers for app store moderators,” pp. 1–11, 2015.
[19] X. Gu and S. Kim, “" what parts of your apps are loved by users?"(t),” in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2015, pp. 760–770.
[20] W. Maalej, Z. Kurtanovi´ c, H. Nabil, and C. Stanik, “On the automatic classification of app reviews,” Requirements Engineering, vol. 21, no. 3, pp. 311–331, 2016.
[21] V. T. Dhinakaran, R. Pulle, N. Ajmeri, and P. K. Murukannaiah, “App review analysis via active learning: reducing supervision effort without compromising classification accuracy,” in 2018 IEEE 26th international requirements engineering conference (RE). IEEE, 2018, pp. 170–181.
[22] H. Guo and M. P. Singh, “Caspar: Extracting and synthesizing user stories of problems from app reviews,” 2020, p. 628–640.
[23] C. Stanik, M. Haering, and W. Maalej, “Classifying multilingual user feedback using traditional machine learning and deep learning,” in 2019 IEEE 27th international requirements engineering conference workshops (REW). IEEE, 2019, pp. 220–226.
[24] N. Aslam, W. Y. Ramay, K. Xia, and N. Sarwar, “Convolutional neural network based classification of app reviews,” IEEE Access, vol. 8, pp. 185619–185628, 2020.
[26] P. R. Henao, J. Fischbach, D. Spies, J. Frattini, and A. Vogelsang, “Transfer learning for mining feature requests and bug reports from tweets and app store reviews,” in 2021 IEEE29thInternational Requirements Engineering Conference Workshops (REW). IEEE, 2021, pp. 80–86.
[27] J. Verma and A. Patel, “Evaluation of unsupervised learning based extractive text summarization technique for large scale review and feedback data,” Indian Journal of Science and Technology, vol. 10, pp. 1–6, 05 2017.
[28] S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, “Want to reduce labeling cost? gpt-3 can help,” arXiv preprint arXiv:2108.13487, 2021.
[29] X. He, Z. Lin, Y. Gong, A. Jin, H. Zhang, C. Lin, J. Jiao, S. M. Yiu, N. Duan, W. Chen et al., “Annollm: Making large language models to be better crowd sourced annotators,” arXiv preprint arXiv:2303.16854, 2023.
[30] R. Zhang, Y. Li, Y. Ma, M. Zhou, and L. Zou, “Llmaaa: Making large language models as active annotators,” arXiv preprint arXiv:2310.19596, 2023.
[31] J. Zhou, W. Du, M. O. F. Rokon, Z. Wang, J. Xu, I. Shah, K.-c. Lee, and M. Wen, “Enhanced e-commerce attribute extraction: Innovating with decorative relation correction and llama 2.0-based annotation,” arXiv preprint arXiv:2312.06684, 2023.
[32] Z. He, C.-Y. Huang, C.-K. C. Ding, S. Rohatgi, and T.-H. K. Huang, “If in a crowdsourced data annotation pipeline, a gpt-4,” in Proceedings of the CHI Con ference on Human Factors in Computing Systems, 2024, pp. 1–25.
[33] A. Wang, J. Morgenstern, and J. P. Dickerson, “Large language models cannot replace human participants because they cannot portray identity groups,” arXiv preprint arXiv:2402.01908, 2024.
[34] Y. Tang, C.-M. Chang, and X. Yang, “Pdf Annotator: A human-llm collaborative multi-modal data annotation tool for pdf-format catalogs,” in Proceedings of the 29th International Conference on Intelligent User Interfaces, 2024, pp. 419–430.
[35] D. Yu, L. Li, H. Su, and M. Fuoli, “Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apology,” International Journal of Corpus Linguistics, 2024.
[36] N.Pangakis, S. Wolken, and N. Fasching, “Automated annotation with generative ai requires validation,” arXiv preprint arXiv:2306.00176, 2023.
[37] M.Imamovic, S. Deilen, D. Glynn, and E. Lapshinova-Koltunski, “Using chatgpt for annotation of attitude within the appraisal theory: Lessons learned,” in Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII), 2024, pp. 112–123.
[38] J. Tan, A. Zhang, X. Zhang, C. Xiao, Z. Ding, Y. Peng, C. Wu, X. Zhu, J. Zhou, and X. Huang, “Large language models for data annotation: A survey,” arXiv preprint arXiv:2402.13446, 2024.
[39] S. Gunathilaka and N. De Silva, “Aspect-based sentiment analysis on mobile application reviews,” in 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2022, pp. 183–188.
[40] S. Kobayashi, “Contextual augmentation: Data augmentation by words with paradigmatic relations,” 2018.
[41] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., “Llama: Open and efficient foundation language models,” arXiv preprint arXiv:2302.13971, 2023.
[42] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bash lykov, S. Batra, P. Bhargava, S. Bhosale et al., “Llama 2: Open foundation and f ine-tuned chat models,” arXiv preprint arXiv:2307.09288, 2023.
[43] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez et al., “Vicuna: An open-source chatbot impressing gpt-4 with 90
[44] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic et al., “Falcon-40b: an open large language model with state-of-the-art performance,” 2023.
[45] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier et al., “Mistral 7b,” arXiv preprint arXiv:2310.06825, 2023.
[46] L. Reiter, “Zephyr,” Journal of Business Finance Librarianship, vol. 18, no. 3, pp. 259–263, 2013.
[47] G. Colavito, F. Lanubile, N. Novielli, and L. Quaranta, “Leveraging gpt-like llms to automate issue labeling,” in 2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR). IEEE, 2024, pp. 469–480.
[48] W. Maalej and H. Nabil, “Bug report, feature request, or simply praise? on automatically classifying app reviews,” in 2015 IEEE 23rd international requirements engineering conference (RE). IEEE, 2015, pp. 116–125.
[49] D. Yu, L. Li, H. Su, and M. Fuoli, “Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis.”
[50] K. Hamilton, L. Longo, and B. Bozic, “Gpt assisted annotation of rhetorical and linguistic features for interpretable propaganda technique detection in news text.” in Companion Proceedings of the ACM on Web Conference 2024, 2024, pp. 1431–1440.
[51] T. Zhang, I. C. Irsan, F. Thung, and D. Lo, “Revisiting sentiment analysis for software engineering in the era of large language models,” arXiv preprint arXiv:2310.11113, 2023.
[52] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Ha jishirzi, “Self-instruct: Aligning language models with self-generated instruc tions,” arXiv preprint arXiv:2212.10560, 2022.
[53] R. R. Mekala, Y. Razeghi, and S. Singh, “Echoprompt: Instructing the model to rephrase queries for improved in-context learning,” 2024. [Online]. Available: https://arxiv.org/abs/2309.10687
[54] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, P. S. Dulepet, S. Vidyadhara, D. Ki, S. Agrawal, C. Pham, G. Kroiz, F. Li, H. Tao, A. Srivastava, H. D. Costa, S. Gupta, M. L. Rogers, I. Goncearenco, G. Sarli, I. Galynker, D. Peskoff, M. Carpuat, J. White, S. Anad kat, A. Hoyle, and P. Resnik, “The prompt report: A systematic survey of prompt ing techniques,” 2024.
[55] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient fine tuning of quantized llms,” 2023. [Online]. Available: https: //arxiv.org/abs/2305.14314
[56] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan, “Peft: State-of-the-art parameter-efficient fine-tuning methods,” https://github. com/huggingface/peft, 2022.

References

[57]. R. V. Krejcie and D. W. Morgan, “Determining sample size for research activities,” Educational and psychological measurement, vol. 30, no. 3, pp. 607–610, 1970.
[58] J. Cohen, “Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit.” Psychological bulletin, vol. 70, no. 4, p. 213, 1968.
[59] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” biometrics, pp. 159–174, 1977. [60] L. Chen, M. Zaharia, and J. Zou, “How is chatgpt’s behavior changing over time?” arXiv preprint arXiv:2307.09009, 2023.

Thank You!


Q
A