# RE-Miner: Mining Mobile User Reviews with Feature Extraction and Emotion Classification

Quim Motger[1], Max Tiessler[1], Marc Oriol[1] and Irene Bertolín[1]

[1]*Department of Service and Information System Engineering, Universitat Politècnica de Catalunya*

Abstract

In the context of app stores, user reviews are pivotal on supporting multiple requirements engineering tasks. Among these, feature extraction and emotion classification play a crucial role in requirements prioritization, feedback gathering and release planning. Empirical evaluation of these techniques is impeded by data collection complexities and a lack of reproducible methods and available tools. Furthermore, existing studies often focus on isolated tasks, hindering a comprehensive analysis of user perceptions. This paper introduces RE-Miner, a work-in-progress tool integrating multiple feature extraction and emotion classification innovative methods, enabling a detailed analysis of feature-oriented user feedback. RE-Miner comprises a web-based service for task integration and comparison, and a web application for persistent storage and analytical visualization of reviews. As a result, RE-Miner provides a platform for seamless integration, replication, and comparison of review mining techniques, fostering advancements in feature extraction and emotion classification understanding for requirements engineering. A demo of the tool is showcased here: https://youtu.be/PFNCbborPuU.

**Keywords**

mobile app reviews, feature extraction, emotion classification, natural language processing

## 1. Introduction

User reviews offer a wealth of information to support multiple requirements engineering tasks [1]. Elicitation of feature requests [2], identification of bugs or issues [3], user feedback gathering and analysis [4], and release planning or prioritization [5] are a few examples of the most popular use cases for automated review processing. Among these, feature extraction (i.e., extracting mentions to functional aspects of an app [6]) and emotion classification (i.e., extracting the sentiments or emotions in a text [7]) are two of the most popular techniques. While they have undergone intense study [8], innovations in the landscape of natural language processing triggered by deep learning and large language models are promoting accuracy improvements in both feature extraction [9] and emotion classification [10]. Meanwhile, some challenges like disambiguation, domain-specific adaptation and precision of negative emotion detection still remain [8].

*In: D. Mendez, A. Moreira, J. Horkoff, T. Weyer, M. Daneva, M. Unterkalmsteiner, S. Bühne, J. Hehn, B. Penzenstadler, N. Condori-Fernández, O. Dieste, R. Guizzardi, K. M. Habibullah, A. Perini, A. Susi, S. Abualhaija, C. Arora, D. Dell'Anna, A. Ferrari, S. Ghanavati, F. Dalpiaz, J. Steghöfer, A. Rachmann, J. Gulden, A. Müller, M. Beck, D. Birkmeier, A. Herrmann, P. Mennig, K. Schneider. Joint Proceedings of REFSQ-2024 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2024. Winterthur, Switzerland, April 8, 2024.*

✉ joaquim.motger@upc.edu (Q. Motger); max.tiessler@upc.edu (M. Tiessler); marc.oriol@upc.edu (M. Oriol)
iD 0000-0002-4896-7515 (Q. Motger); 0000-0003-1928-7024 (M. Oriol)

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Evaluating these methods poses scientific challenges like data collection [11], replication and deployment of resource-intensive services [12]. Additionally, from a user perspective, comparison and selection of the most suitable approach for a given domain is problematic, undermined or even neglected [13]. Furthermore, most research is dedicated to the isolated use of these tasks [8]. Combining feature and emotion descriptors allows a fine-granularity analysis on the user perception to a particular feature or a cluster of features. This knowledge is valuable to support single-feature emotion-oriented analysis and filtering of non-relevant content [14]. While there is some existing work proposing a combined analysis of these tasks (see Section 6), their replication is problematic due to the lack of open source tools and reusable frameworks [8].

This paper introduces **RE-Miner**, a work-in-progress software tool designed for replication and comparative analysis in review-based feature extraction and emotion classification. RE-Miner consists of a web-based service for integrating and comparing multiple review mining tasks, and a web application to support the visual analysis of user reviews, incorporating statistical data on the features and emotions derived from these reviews. We envisage that our contribution will assist researchers and app stakeholders in the selection, replication, integration and comparison of review mining techniques.

## 2. Background

While the field of natural language processing for requirements engineering (NLP4RE) is not novel, availability of reusable tools or even full descriptions to allow replication of such techniques is scarce. Zhao et al. surveyed 130 NLP4RE tools [13], from which only 17 were available for download. Furthermore, they claimed this scarcity to be particularly highlighted in novel NLP techniques and specialized tools integrating deep learning strategies.

Focusing on feature extraction tasks, the SAFE tool is considered one of the standard methods, based on syntactic-based pattern matching techniques complemented with semantic similarity for feature candidate linking [6]. However, not only its accuracy in the analysis of reviews is limited [15], but its source code is not available and its replication requires design assumptions [8]. Similar approaches like GuMa [14] and ReUS [16] also focus on syntactic aspects. Beyond this formulation of the problem, KEFE proposes a deep learning classifier designed to sift through syntactically extracted features and exclude non-relevant expressions [17].

On the other hand, emotion classification aims at detecting the emotion in accordance with a specified emotional model. This contrasts to traditional sentiment analysis techniques that just aim at measuring the positive or negative orientation of a text [18]. In our proposal, we have adopted one of the most widely embraced emotion models, proposed by P. Ekman [19]. This model classifies the emotions as: sadness, fear, happiness, anger, surprise, and disgust. Complementary, the neutral feeling is also used for non-specific emotions. Automatic emotion identification methods from textual content include lexicon-based techniques, classical machine learning models (e.g. SVM, bayes,...) or deep learning models [18]. More recently, transformer-based models and LLMs have significantly advanced the field of emotion classification [20].

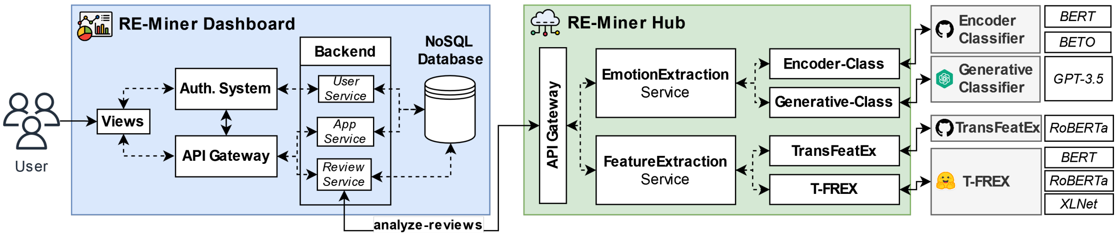


**Figure 1:** RE-Miner architecture. External service integration is represented: (1) as a web-service built from source code used through API communication; (2) as a model loaded from HuggingFace for inference; or (3) as a fine-tuned model from OpenAI used through API communication.

## 3. Tool description

The main goal of RE-Miner is to provide a user-friendly, easily accessible and reusable software tool for both researchers and mobile app developers. It is designed to facilitate replication studies and comparative analyses in the field of review-based natural language processing tasks, with a particular emphasis on feature extraction and emotion classification. The tool is composed of two software-based contributions: (1) RE-Miner-Hub[1], a web-based service supporting the integration and comparison of feature extraction and emotion classification tasks, using various NLP models and providers; and (2) RE-Miner-Dashboard[2], a web-based application to support user management, app and review persistence, automatic processing and analytical visualization of a batch of user reviews combining features and emotions emerged from these reviews. Figure 1 provides a high-level overview of RE-Miner architecture.

### 3.1. RE-Miner-Hub

RE-Miner-Hub is a web-service system designed to empower researchers with the capability to conduct feature extraction and emotion classification tasks using a common API syntax. RE-Miner-Hub serves as a centralized orchestration service of heterogeneous software components (both from a logic and physical point of view), each of them deployed as decoupled, decentralized software resources. This architecture facilitates re-usability of third-party methods, which can extend RE-Miner set of tasks by either replicating and embedding these techniques as a new RE-Miner software module or simply by using available services from the web. For feature extraction, current version of RE-Miner-Hub employs two methods based on our previous work:

- **TransFeatEx** integrates a RoBERTa-based pre-trained model used to leverage syntactic patterns and semantic annotations (e.g., polarity score) to identify feature expressions [21]. TransFeatEx is developed as a Python-based web-service, and it is deployed as a standalone web application and accessed using HTTP-based communication through an API. We use its default configuration, as described in the tool repository and in the original paper [21].
- **T-FREX** gives access to a suite of Transformer-based models (i.e., BERT, RoBERTa and XLNet) fine-tuned for Named-Entity Recognition using crowdsourced feature annotations

[1] Source code and API Swagger documentation available at: https://github.com/gessi-chatbots/RE-Miner-Hub.
[2] Source code and sample dataset available at: https://github.com/gessi-chatbots/RE-Miner-Dashboard.

generated by users in software recommendation platforms [9]. T-FREX models are available on HuggingFace[3], allowing their integration by simply adding these models for inference through an NLP pipeline using the Transformers Python module.

For emotion classification, we developed and employed the following methods:

- **Encoder Classifier** focuses on fine-tuning encoder-only LLMs (i.e., BERT, BETO) with a document classification layer on top [22]. These models were fine-tuned using 2,000 annotated user-generated microblogs (e.g., tweets). This component weights each emotion class, based on the probability distribution of a given review to reflect said property.
- **Generative Classifier** utilizes a GPT-3.5-Turbo model fine-tuned on a few-shot learning setting using prompt engineering to extract emotions from mobile application reviews. We used a reduced sample dataset of 100 internally annotated app reviews for the fine-tuning process[4]. Contrarily to the Encoder Classifier, the outcome of this method is restricted to the most probable emotion for a given review.

For each task, users have the freedom to choose models that align with their preferences (i.e., context, dataset, computational resources...). RE-Miner-Hub exposes simple API methods to perform both feature extraction and/or emotion classification on a batch of reviews.

The modular design of RE-Miner-Hub provides flexibility in the list of available models, enabling scalable integration of new models for both sentiment analysis and feature extraction tasks. Moreover, within the RE-Miner ecosystem, a unified data model for reviews is used. This ensures scalability when integrating other software components that operate with the same data model. Lastly, the Hub acts as a flexible middle-ware between the RE-Miner-Dashboard and various third-party APIs and software components. The only requirement for integrating a new model into the RE-Miner-Hub is that it must be accessible via API (either the Hugging Face inference API or a traditional REST API).

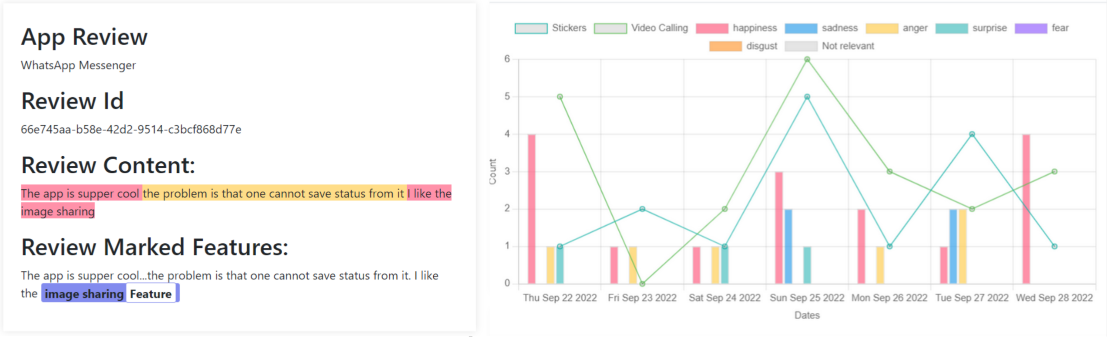
### 3.2. RE-Miner-Dashboard

The RE-Miner-Dashboard is primarily designed as a visualization and analytical software component. The dashboard, encompasses several key components: (1) a React front-end application; (2) an authorization and authentication system; (3) an API Gateway responsible for managing the access to the APIs; (4) a backend consisting of two APIs (handling reviews and mobile apps, respectively) and a module dedicated to creating new user entities within the database; and (5) a NoSQL document-based database.

Upon user creation and the corresponding database entry, access permissions to the application and associated APIs are granted. Users can upload individual applications or batches of them, along with the reviews, which are then stored in the database. When users want to analyze a review or a batch of reviews, the RE-Miner-Dashboard application sends a request to the RE-Miner-Hub, specifying the task (e.g. feature extraction and/or sentiment analysis). When receiving results from the RE-Miner-Hub system, they are stored in the database. This

[3] E.g.: https://huggingface.co/quim-motger/t-frex-bert-base-uncased (T-FREX models are referenced in model card).

[4] Prompt is available in GitHub repository. Full evaluation is yet to be conducted as depicted in Section 5.


App Review

WhatsApp Messenger

Review Id

66e745aa-b58e-42d2-9514-c3bcf868d77e

Review Content:

The app is supper cool the problem is that one cannot save status from it I like the image sharing

Review Marked Features:

The app is supper cool...the problem is that one cannot save status from it. I like the image sharing Feature

**Figure 2:** RE-Miner-Dashboard, including review analysis (left) and features/emotions chart (right).

seamless integration empowers users to use the analytical dashboard for engaging visual analyses of their data, extracting meaningful insights from the reviews. Details on the use cases and RE-Miner-Dashboard views are showcased in Section 4.

## 4. User workflow

The RE-Miner-Dashboard presents two main use cases for review analysis. These require users to have completed the sign-up, login and upload of apps and reviews. Figure 2 illustrates a snapshot of both use cases.

1. **Single-review analysis**
   a) Users can select a review for analysis from the *Reviews > View Reviews* tab. Initiation of the analysis process can be done by clicking the *Process review* button.
   b) A modal wizard appears, guiding the user through the following steps:
      i. The user selects the task/s and the method used for each task (e.g., GPT-3.5 for emotion classification; T-FREX BERT base for feature extraction).
      ii. The reviews are submitted to the RE-Miner-Hub to initiate the review analysis.
   c) After analysis, an icon 👁 will appear next to the processed review. By clicking it, the user opens the *Review Analyzer* view, displaying the analysis results, including detected emotions, emotion-marked sentences, and identified features within the review text.
2. **Batch-review analysis and visual analytics**
   a) The user should repeat steps *1.a)* and *1.b)* while selecting multiple (or all) uploaded app reviews.
   b) By navigating to the *Dashboard* tab, the following analytical charts can be found:
      i. *Sentiment Polar Area*: aggregated sum of each emotion across all reviews.
      ii. *Top Features Histogram*: aggregated sum of the most frequent extracted features.
      iii. *Features Over Time Chart*: distribution of feature mentions over a time window.
      iv. *Sentiment Histogram*: distribution of emotions (displayed in a stacked layout based on frequency for each emotion class) over a time window.
      v. *Features/Emotions Chart*: combined distribution of a set of feature mentions with their associated emotions over a time window.

## 5. Evaluation plan

Below we summarize the main steps of the planned (and ongoing) evaluation:

- **Data collection and annotation.** We built on our previous work on mobile app repository mining to collect multiple reviews for a given domain [23], filtered and refined for the feature extraction task [9]. This dataset consists of 468 apps with 23,816 reviews within a 1 year time window, each with at least 1 crowdsourced annotated feature. We used a subset of this data set to showcase the different use cases of RE-Miner as depicted in Section 4 and in the video demonstration. For emotion classification, we plan on conducting an internal iterative annotation process of a subset of reviews through structured guidelines, measuring annotation agreement and establishing solid evaluation criteria.
- **Experimentation.** We plan to conduct an empirical evaluation of all methods in Section 3.1 by combining multiple cross-validation analyses using the complete data set annotated with features and emotions. This entails quantitative ground-truth evaluation to assess and compare the effectiveness of each technique. Additionally, we intend to apply search-based algorithms (e.g., clustering) to infer how the aggregated analysis of features and emotions can support requirements engineering tasks. For assessing the overall software product quality of the tool, we will focus on performance efficiency and usability as defined in ISO/IEC 25010 [24]. For performance efficiency, we will measure the tool's response time and scalability under varying workloads for data upload and feature and emotion extraction tasks. For usability assessment, we plan to conduct user studies to evaluate ease of interaction, explainability of results, and overall user satisfaction with potential stakeholders.

## 6. Related work

Dąbrowski et al. recently concluded that automated combined analysis of feature extraction and emotion classification is limited, especially in the use of innovative NLP techniques [1]. Guzman and Maleej described a syntactic and semantic based technique combining feature extraction with lexicon-based polarity extraction on a sentence-level [14]. This process is consolidated through an LDA topic modelling method to infer high-level features and average sentiments (i.e., positive vs. negative). A similar approach is depicted by Dragoni et al. [16], depicting a pipeline for the streamlined automated collection and analysis of user reviews to extract a normalized polarity score. Beyond methods surveyed by Dąbrowski et al. [1], Gunaratman et al. propose an automated app rating mechanism by weighting features and associated sentiments on a feature level [25]. Finally, TransFeatEx integrates a sentiment analysis filter, whose use is limited to filtering out extremely polarized reviews to avoid biased representation [21].

In comparison with RE-Miner, main limitations of these approaches include: (1) unavailability of full source code or distributed software; (2) limitations for full replication; (3) use of traditional NLP techniques with respect to innovative methods; and (4) lack of data analytics visualization beyond finer granularity analysis on a review level. To overcome these limitations, RE-Miner software components are distributed including source code and packaged web services. README files include instructions to install, deploy and integrate these components, as well as a sample dataset to replicate the demo depicted in Section 4. Finally, while RE-Miner does not exclude the use of traditional NLP methods, the current version integrates multiple Transformer-based approaches for both feature extraction and sentiment analysis tasks.

## 7. Conclusions and future work

RE-Miner contributions are three-fold. First, we aim at providing a reusable tool to facilitate integration and extension of NLP4RE tasks in the context of app review mining. Second, we distribute software components as source code and as a standalone web application to integrate and run review mining processes to analyze the output of these methods on a fine granularity (i.e., review and sentence level) basis. Finally, we introduce a sample of simple data analytics to support and evolve the evolution of a dashboard to visualize statistics on review descriptors (i.e., features and emotions), both in isolation and combined.

As ongoing future work, we are working on extending the analytical dashboard with clustering and topic modelling techniques to provide higher levels of abstraction of clusters of features and emotions. This visualization will be used to support user trend analysis, involving dense centroids associated to a particular emotion or set of features. As a next action point, we plan to extend current tasks by including content classification of these reviews as an additional descriptor, focused on topic modelling and type of review (e.g., feature request, bug report, praise...). Finally, from a maintainability perspective, we plan on extending each task with new methods in the field, to facilitate replication studies following open-science principles.

## Acknowledgments

With the support from the Secretariat for Universities and Research of the Ministry of Business and Knowledge of the Government of Catalonia and the European Social Fund. This paper has been funded by the Spanish Ministerio de Ciencia e Innovación under project / funding scheme PID2020-117191RB-I00 / AEI/10.13039/501100011033.

## References

[1] J. Dąbrowski, et al., Analysing app reviews for software engineering: a systematic literature review, Empirical Softw. Engg. 27 (2022).

[2] C. Iacob, R. Harrison, Retrieving and analyzing mobile apps feature requests from online reviews, in: Int. Working Conference on Mining Software Repositories, 2013.

[3] W. Maalej, H. Nabil, Bug report, feature request, or simply praise? On automatically classifying app reviews, in: Int. Requirements Engineering Conference, 2015.

[4] D. Pagano, W. Maalej, User feedback in the appstore: An empirical study, in: Int. Requirements Engineering Conference, 2013.

[5] L. Villarroel, et al., Release planning of mobile apps based on user reviews, in: Proceedings - Int. Conference on Software Engineering, 2016.

[6] T. Johann, et al., SAFE: A Simple Approach for Feature Extraction from App Descriptions and App Reviews, in: Int. Requirements Engineering Conference, 2017.

[7] B. Lin, et al., Sentiment analysis for software engineering: How far can we go?, in: Int. Conf. of Software Engineering, 2018.

[8] J. Dąbrowski, et al., Mining and searching app reviews for requirements engineering: Evaluation and replication studies, Information Systems 114 (2023).

[9] Q. Motger, et al., T-FREX: A Transformer-based Feature Extraction Method from Mobile App Reviews, in: Conference on Software Analysis, Evolution and Reengineering, 2024.

[10] S. L. Ramaswamy, J. Chinnappan, RecogNet-LSTM+CNN: a hybrid network with attention mechanism for aspect categorization and sentiment classification, Intel. Information Systems 58 (2022).

[11] F. Palomba, et al., Crowdsourcing user reviews to support the evolution of mobile apps, Journal of Systems and Software 137 (2018).

[12] R. Jongeling, P. Sarkar, S. Datta, A. Serebrenik, On negative results when using sentiment analysis tools for software engineering research, Empirical Software Engineering 22 (2017).

[13] L. Zhao, et al., Natural language processing for requirements engineering: A systematic mapping study, ACM Comput. Surv. 54 (2021).

[14] E. Guzman, W. Maalej, How do users like this feature? A fine grained sentiment analysis of App reviews, in: Int. Requirements Engineering Conference, 2014.

[15] F. A. Shah, et al., Is the SAFE Approach Too Simple for App Feature Extraction? A Replication Study, Lecture Notes in Computer Science (2019).

[16] M. Dragoni, M. Federici, A. Rexha, An unsupervised aspect extraction strategy for monitoring real-time reviews stream, Information Processing & Management 56 (2019).

[17] H. Wu, et al., Identifying key features from app user reviews, in: ICSE, 2021.

[18] P. Nandwani, R. Verma, A review on sentiment analysis and emotion detection from text, Social Network Analysis and Mining 11 (2021).

[19] P. Ekman, Basic Emotions, John Wiley & Sons, Ltd, 1999.

[20] D. Carneros-Prado, et al., Comparative Study of Large Language Models as Emotion and Sentiment Analysis Systems: A Case-Specific Analysis of GPT vs. IBM Watson, in: Int. Conference on Ubiquitous Computing & Ambient Intelligence, 2023.

[21] A. Gállego, Q. Motger, X. Franch, J. Marco, TransFeatEx: a NLP pipeline for feature extraction, in: Joint proceedings of REFSQ-2023, CEUR-WS.org, 2023.

[22] A. d. Arriba, M. Oriol, X. Franch, Applying transfer learning to sentiment analysis in social media, in: Int. Requirements Engineering Conference Workshops (REW), 2021.

[23] Q. Motger, et al., Mobile feature-oriented knowledge base generation using knowledge graphs, in: New Trends in Advanced Database and Information Systems, 2023.

[24] ISO/IEC, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models ISO/IEC 25010, Technical Report, 2011. URL: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010.

[25] I. Gunaratnam, D. Wickramarachchi, Computational model for rating mobile applications based on feature extraction, in: Int. Conference on Advancements in Computing, 2020.