Journal of Computer Science

Original Research Paper

Leveraging Transformer Model for Data-Driven Requirement Engineering: A Case Study of NLP4RE E-Commerce Business Apps in Malaysia and Indonesia

[1,2]Devi Yurisca Bernanda, [1]Dayang Norhayati Abang Jawawi, [1]Shahliza Abd Halim, [2]Fransiskus Adikara and [2]Cevi Herdian

[1]*Department of Computer Science, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia*
[2]*Department of Data Science, Faculty of Technology and Design, Universitas Bunda Mulia, Jakarta, Indonesia*

Article history
Received: 05-08-2024
Revised: 25-08-2024
Accepted: 27-08-2024

Corresponding Author:
Devi Yurisca Bernanda
Department of Computer Science, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
Email: bernanda@graduate.utm.my

Abstract: Data-Driven Requirement Engineering (DDRE) marks a departure from conventional methods, embracing dynamic, user-centric strategies with the help of machine learning and artificial intelligence. As software systems grow more intricate and demand adaptability, a continuous software engineering approach becomes crucial. However, this shift poses challenges in handling data inconsistencies during the elicitation process, leading to software development shortcomings. This study tackles these challenges by utilizing innovative techniques like Natural Language Processing (NLP) for elicitation, web scraping for feedback collection from platforms like the Google Play Store, and transformer models for sentiment analysis and classification. By collecting data from user-generated feedback and ensuring compatibility, the research aims to streamline requirements gathering. In summary, this research highlights the significance of data-driven approaches in meeting evolving stakeholder needs and organizational objectives priority in functional and non-functional requirement engineering, fostering more effective and accurate requirement engineering practices.

Keywords: DDRE, NLP, Transformer Model, Sentiment Analysis, Classification

Introduction

In the rapidly evolving landscape of software development (Chakir *et al.*, 2024) Requirement Engineering (RE) continues to be a crucial stage that impacts the success and productivity of projects (Adetoba and Ogundele, 2018; Maalej *et al.*, 2019) with businesses heavily depending on software systems or apps, especially in fast-moving industries like e-commerce (Christianto *et al.*, 2023) the necessity for accurate and thorough requirement collection is more critical than ever (Ghasemi, 2018).

Table (1) explains that traditional requirement engineering techniques such as questionnaires, document analysis, or prototyping despite being essential (Altarturi *et al.*, 2017), frequently fall short in managing the vast and complex data produced in the modern digital landscape the difference between Traditional RE and DDRE (Bergrahm and Johansson, 2020).

Table 1: Comparing traditional RE and DDRE (Maalej *et al.*, 2019)

Aspect	Traditional RE	DDRE
Process approach	Follows a step-by-step, linear methodology with distinct phases (elicitation, analysis, specification validation, management)	Adopts an iterative and agile method, promoting of ongoing refinement and adjustment of requirements
Documentation	Focuses on, comprehensive formal requirements specification documents	Relies on data-focused methods with dashboards and dynamic reports, placing less emphasis on formal documentation
Stakeholder involvement	Depends heavily on direct interaction with stakeholders through interviews, surveys, meetings	On indirect and stakeholder engagement by examining the user behavior, feedback, and operational data
Flexibility	Less adaptable; changes are managed through a formal, often slow, change control process	Highly adaptable; to permit real-time or near-real-time adjustments based on data trends and feedback

DOI: 10.3844/jcssp.2024.1504.1509

The comparative analysis in the table highlights several compelling reasons why individuals and organizations prefer DDRE over traditional RE (Franch, 2021). These reasons include increased flexibility, the ability to provide real-time feedback, reliance on data for decision-making, better adaptability to changing requirements, and a more comprehensive approach to stakeholder involvement (Perini, 2019).

To enhance the requirements elicitation process and address contemporary challenges (Adikara *et al*., 2016) the DDRE framework employs various advanced methods such as machine learning and NLP (Perini, 2019). Machine learning method to make using machine learning to make predictive analytics (Vogelsang and Borg, 2019). NLP Analyzes textual data from various sources like user feedback and social media to extract meaningful requirements, and sentiment (Nagpal *et al*., 2022). As a result, adopting DDRE methodologies enables a more dynamic and accurate requirements engineering process, aligning development efforts more closely with user needs and market demands (Nayebi, 2018).

One of those methods is the transformer model for NLP is a type of deep learning model mainly used in Natural Language Processing for Requirement Engineering (NLP4RE), have shown exceptional success in multiple applications due to their high accuracy in comprehending and generating human language (Rahali and Akhloufi, 2023). Incorporating these models into the requirement engineering process allows for the automation and improvement of the extraction, analysis, and validation of requirements from user feedback (Patwardhan *et al*., 2023).

Materials

Dataset

The dataset used in this study was obtained from the google play store. The data consists of app reviews. On this study researcher collected e-commerce shopee from Malaysia and Indonesia:

- Shopee Malaysia: com.shopee.my
- Shopee Indonesia: com.shopee.id

Data Collection

Data was collected using the google-play-scraper Python library, which provides an easy interface to extract reviews and other app-related information from the google play store. To begin using the google-play-scraper library, first, install it using pip by running pip install google-play-scraper. After installation, import the necessary modules to access app reviews, such as app and reviews all. The app function allows you to collect basic information about an app, including its title, developer, rating, and other metadata. To gather comprehensive user feedback, the reviews all function can be used to retrieve all reviews of a specific app, providing detailed insights such as user comments, ratings, and the date of each review.

Methods

Figure (1) explains how to investigate the use of the Transformer model in DDRE for e-commerce businesses like shopee in Malaysia and Indonesia, we implemented the following steps involving four main steps: (1) Web scraping user feedback, (2) Translating into English if non-English language, (3) Conducting positive and negative sentiment analysis, (4) Requirements into functional and non-functional, and sub-classifying these requirements into several subclass (Pichiyan *et al*., 2023).

Web Scraping

In this research, the google play store scraper python library was chosen for web scraping. The google play store scraper is highly favored for extracting data from google play.

Store due to its specialized design tailored specifically to the platform's needs.

The web scraping process yielded a CSV dataset containing 796 rows, collected within the timeframe from 5/12/2024, 2:18:00 to 5/11/2024, 6:34:00 PM. Due to the limitations of our computer's processing capacity, we will restrict the number of rows used in subsequent analysis.

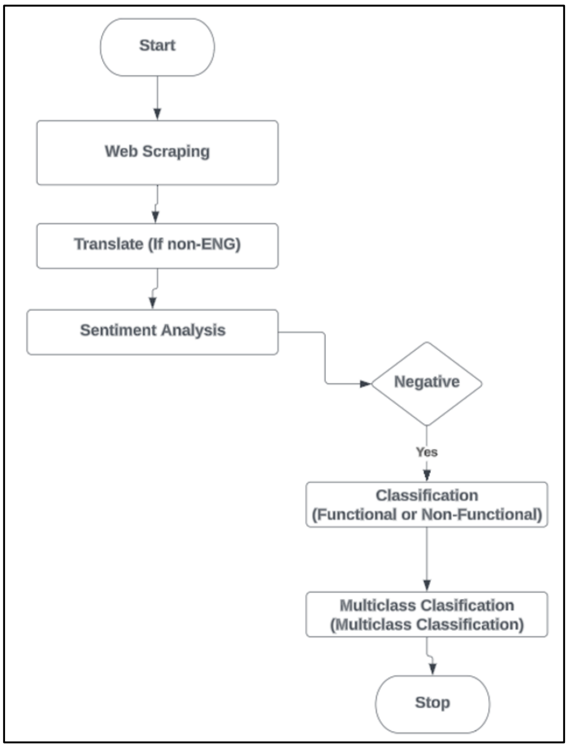

Fig. 1: Flowchart research

DOI: 10.3844/jcssp.2024.1504.1509

Translating into English

Given that the review texts are in non-English languages; it is necessary to translate them into English to facilitate our analysis. To achieve this, we also utilized a Transformer model for the translation process, ensuring accurate and contextually appropriate conversions of the original text (Rahali and Akhloufi, 2023). As our study focuses on shopee operations in Malaysia and Indonesia, it's imperative that we translate the data into English for comprehensive analysis (ID to ENG and MY to ENG).

Sentiment Analysis: Positive Negative

Sentiment analysis, also referred to as opinion mining, constitutes a Natural Language Processing (NLP) method employed to scrutinize and ascertain the sentiment or emotional tone conveyed in a text. The objective of sentiment analysis lies in grasping the prevailing sentiment conveyed through the text, be it positive or negative (Rahali and Akhloufi, 2023).

Classification: Functional Requirement or Non-Functional Requirement

In NLP, classification involves assigning text data to predefined categories or classes according to their content or characteristics. This process entails training a machine learning model to anticipate the category or class of new text samples using patterns gleaned from labeled training data (Rahali and Akhloufi, 2023).

Multiclass Classification

In software development, functional requirements delineate the actions a system ought to undertake, encompassing distinct features, functionalities, and behaviors it should demonstrate. Conversely, non-functional requirements detail the manner in which the system operates, covering facets such as performance, security, reliability, usability, and scalability. While functional requirements concentrate on the system's actions, non-functional requirements emphasize the system's effectiveness in executing those actions. For multiclass classification, we will still use the transformer model. Unlike binary classification, which distinguishes between functional and non-functional categories, multiclass classification requires inputting all possible labels (Ahmed *et al*., 2022).

Results

The research workflow began with web scraping (Fig. 2), where data was collected from the google play store using the google-play-scraper library. This step involved extracting user reviews, ratings, and other relevant information from various mobile applications. The collected data, originally in multiple languages (Indonesia and Malaysia), was then translated into English using a machine translation tool (Fig. 3). This translation process ensured that all reviews could be uniformly processed for subsequent sentiment analysis, regardless of the original language, thus broadening the scope and inclusivity of the dataset.

Following translation, the sentiment analysis was performed (Fig. 4) to categorize the user reviews into positive or negative sentiments. This analysis provided a foundational understanding of user perceptions and feedback, which is crucial for identifying the general mood and reception of the app features. The sentiment-labeled data was then further classified into functional and non-functional requirements (Fig. 5), helping to distinguish between comments related to the core functionality of the application (e.g., features and performance) and those related to broader aspects such as user experience or design.

Finally, a multiclass classification was conducted (Fig. 6) to categorize the reviews into more specific requirements. This step enabled the identification of detailed aspects of app performance, such as usability, security, or compatibility, providing a granular view of user feedback. This comprehensive workflow allowed for a nuanced understanding of user sentiments and requirements, facilitating more targeted improvements to app development and user satisfaction.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	source	review_id	user_nam	review_tit	review_de	rating	thumbs_u	review_da	developer	developer	appVersio	laguage_c	country_code	
0	Google Pla	816d9c3a-	Aziq Firdauz		Nice	4	0	########	We're hap	########	3.24.17	id	id	
1	Google Pla	6fe7c3b5-	Muhammad Iqbal		Oke	5	0	########	Kami	########		id	id	
2	Google Pla	09e726b8-	Bhabun Niam		ok	5	0	########	We're hap	########	3.24.17	id	id	
3	Google Pla	efa39124-	Albertus Bria		Yang teba	5	0	########	Kami	########	3.24.17	id	id	
4	Google Pla	84494fb7-	Bujang Lasak		Best of the	5	0	########	We're hap	########	3.24.17	id	id	
5	Google Pla	4224edec-	Sulisna Santi		Sangat ba	5	0	########	Kami	########	3.24.17	id	id	
6	Google Pla	1c14df84-	Achmad Nadif		Puas hati	3	0	########	Kami	########	3.25.11	id	id	

Fig. 2: CSV file after scraping

```python
from transformers import M2M100ForConditionalGeneration, M2M100Tokenizer

# Input Malay text
malay_text = "Hidup itu seperti sebuah kotak coklat."

# Load the model and tokenizer
model = M2M100ForConditionalGeneration.from_pretrained("facebook/m2m100_418M")
tokenizer = M2M100Tokenizer.from_pretrained("facebook/m2m100_418M")

# Set source and target languages
tokenizer.src_lang = "ms"  # Language code for Malay
tokenizer.tgt_lang = "en"  # Language code for English
```

Fig. 3: Translating MY into ENG

```python
import torch
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification

tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")
model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

inputs = tokenizer("The system is error", return_tensors="pt")
with torch.no_grad():
    logits = model(**inputs).logits

predicted_class_id = logits.argmax().item()
model.config.id2label[predicted_class_id]
```

Fig. 4: Transformer model distil BERT for sentiment analysis

DOI: 10.3844/jcssp.2024.1504.1509

```
from transformers import pipeline
classifier = pipeline("zero-shot-classification",
                      model="facebook/bart-large-mnli")

sequence_to_classify = "Choose one-hour delivery after the transaction, but it's been four hours and
still not shipped... The phone number isn't being answered no matter how long you wait."
candidate_labels = ['functional requirements', 'non-functional requirements']
classifier(sequence_to_classify, candidate_labels)
```

Fig. 5: Transformers model for binary classification

```
from transformers import pipeline
classifier = pipeline("zero-shot-classification",
                      model="facebook/bart-large-mnli")

sequence_to_classify = "Choose one-hour delivery after the transaction, but it's been four hours and
still not shipped... The phone number isn't being answered no matter how long you wait."
candidate_labels = ['Performance', 'Security', 'Reliability', 'Scalability', 'Usability',
'Compatibility', 'Accessibility', 'Data integrity', 'Error handling', 'Regulatory compliance']
classifier(sequence_to_classify, candidate_labels)
```

Fig. 6: Multiclass classification using transformers model

Table 2: NLP example of shopee MY and ID

	Shopee my	Shopee ID
Review	Sngt teruk, shopee bekukan akun sye untuk claim voucher free shipping no min spend. sye hnya di benarkan guna voucher free shipping dengan minimal order Rm 15. bnyak voucher yang di had kan dari akun sye	Shopee...penjualnya banyak yg penipu.. pesen handphone di kirim sendal.jepit.. pesan timbangan digital chargernya gk.ada.... payaaah
Translate into ENG	Sngt bad, shopee freezing account sye for claim voucher free shipping no min spend. sye hnya is allowed to use voucher free shipping with minimal order Rm 15	Shopee...sales a lot of cheating.. mobile tickets in sendal.jepit..digital mailing..chargernya gk.ada....payaah
Sentiment analysis	Negative: 0.999, positive: 0.001	Negative: 0.997, positive: 0.003
Classification into functional and functional requirements	Functional requirements: 0.60, non-functional requirements: 0.40	Non-functional requirements: 0.51, functional requirements: 0.49
Multiclass classification	Security: 0.189157, shipping: 0.147295	Response time: 0.070833 testability: 0.067691

Table (2) explains that after conducting web scraping, translation, sentiment analysis, classification, and multiclass classification, we will integrate all these processes into a single workflow. This unified workflow will help us determine which functional or non-functional requirements should be prioritized. Establishing this priority will streamline the process, ensuring that the most critical tasks for quality improvement are addressed first.

Discussion

This research describes a full workflow for evaluating user feedback on mobile applications, including web scraping, translation, sentiment analysis, requirement categorization, and multiclass classification. We were able to successfully collect a sizable dataset of customer evaluations from the google play store by starting the process with web scraping. This stage allowed us to record a range of thoughts and experiences linked with various applications, in addition to providing a lot of information and highlighting the varying perspectives of users.

An essential step in ensuring inclusion and accuracy in the analysis was translating the reviews into English. This procedure reduced the possibility of biases resulting from language limitations, allowing a wider variety of data to be handled consistently. We increased the dataset's consistency by using efficient machine translation technologies, which is necessary for accurate sentiment analysis. Reviewers' opinions were further clarified by the sentiment analysis stage, which divided reviews into favorable and unfavorable attitudes. The apps' areas of strength and areas for development were determined using this binary classification as a starting point.

Every stage of the workflow is essential in the data processing environment for converting unprocessed user input into useful insights. Sentiment analysis and categorization phases could be greatly improved by incorporating cutting-edge methods like machine learning algorithms and Natural Language Processing (NLP). Applying NLP approaches, for example, could enhance the extraction of contextual sentiments and nuanced meanings from user evaluations, enabling more precise sentiment classification.

Furthermore, using advanced machine learning models for multiclass classification may result in more accurate classifications of user input, which could eventually provide a more thorough grasp of user needs. Furthermore, the translation and sentiment analysis stages of the data processing pipeline could be streamlined by implementing automated data cleaning techniques. We can make sure that the insights from user feedback successfully contribute to app development strategies and boost user happiness in the quickly changing mobile app landscape by iteratively improving this process and implementing these enhancements.

Conclusion

To gain a comprehensive understanding, analyzing a single piece of feedback is insufficient; instead, it is necessary to aggregate all feedback data collected. This involves applying the steps to the entire dataset. Furthermore, the feedback dataset is continuously growing, necessitating the development of a dynamic dashboard to monitor changes in these requirements over time. This approach ensures that we stay updated with evolving customer needs and preferences, thereby facilitating more informed decision-making and strategy formulation. But for doing that there are limitations like computer power and time-consuming.

Acknowledgment

The manuscript's author expresses appreciation and gratitude to Universiti Teknologi Malaysia and Universitas Bunda Mulia for their support in this research.

Funding Information

The researchers would like to thank the directorate of the center for research and community service (P3M) Universitas Bunda Mulia for funding this research.

Author's Contributions

Devi Yurisca Bernanda: Conceptualized the research problem, interpreted the result, and written the manuscript.

Dayang Norhayati Abang Jawawi: Define the research questions and obtain the data for this submission.

Shahliza Abd Halim: Established work plans and concept design.

Fransiskus Adikara: Explaining the attributes and data types of the source data.

Cevi Herdian: Preparing the data for analysis, training, and implementation.

Ethics

This article is original and unpublished. Correspondence author confirms that all other authors have read and agree that the manuscript does not involve ethical issues.

References

Adetoba, B., & Ogundele, I. (2018). Requirements Engineering Techniques in Software Development Life-Cycle Methods: A Systematice Literature Review. *International Journal of Advanced Research in Computer Engineering & Technology*, *7*(10), 733–743.

Adikara, F., Hendradjaya, B., & Sitohang, B. (2016). Organization Goal-Oriented Requirements Elicitation Process to Enhance Information System. *International Journal of Electrical and Computer Engineering (IJECE)*, *6*(6), 3188–3195. https://doi.org/10.11591/ijece.v6i6.12802

Ahmed, A. K., Zaied, A. N. H., & Hagag, R. (2022). Using Data Mining Techniques to improving Key Performance Indicators. *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 63–68. https://doi.org/10.1109/miucc55081.2022.9781743

Altarturi, H. H., Ng, K.-Y., Ninggal, M. I. H., Nazri, A. S. A., & Ghani, A. A. A. (2017). A Requirement Engineering Model for Big Data Software. *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, 111–117. https://doi.org/10.1109/icbdaa.2017.8284116

Bergrahm, A., & Johansson, O. (2020). *Data-Driven Decision Making in Require-ments Engineering*. Chalmers Reproservice.

Chakir, A., Andry, J. F., Bansal, R., Ulah, A., Durrah, O., & Mamoon, R. (2024). Using Big Data and Artificial Intelligence to Enhance Smart Cities. In *Leveraging ChatGPT and Artificial Intelligence for Effective Customer Engagement* (pp. 214–232). IGI Global. https://doi.org/10.4018/979-8-3693-0815-8.ch013

Christianto, K., Bernanda, D. Y., Lee, F. S., Andry, J. F., & Karepowan, N. N. (2023). Testing Quality Web-Application E-commerce BookCorner Using ISO 9126. *International Journal of Open Information Technologies*, *11*(7), 122–127.

Franch, X. (2021). Data-Driven Requirements Engineering: A Guided Tour. *Communications in Computer and Information Science*, 83–105. https://doi.org/10.1007/978-3-030-70006-5_4

Ghasemi, M. (2018). What Requirements Engineering can Learn from Process Mining. *2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE)*, 8–11. https://doi.org/10.1109/d4re.2018.00008

Maalej, W., Nayebi, M., & Ruhe, G. (2019). Data-Driven Requirements Engineering - An Update. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 289–290. https://doi.org/10.1109/icse-seip.2019.00041

Nagpal, D., Malik, K., & Kalia, A. (2022). A Comprehensive Analysis of Requirement and Design Using Machine Learning. *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022*. Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022. https://doi.org/10.2139/ssrn.4034866

Nayebi, M. (2018). Data Driven Requirements Engineering: Implications for the Community. *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 439–441. https://doi.org/10.1109/re.2018.00058

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the Real World: A Survey on NLP Applications. *Information*, *14*(4), 242. https://doi.org/10.3390/info14040242

DOI: 10.3844/jcssp.2024.1504.1509

Pichiyan, V., Muthulingam, S., G, S., Nalajala, S., Ch, A., & Das, M. N. (2023). Web Scraping Using Natural Language Processing: Exploiting Unstructured Text for Data Extraction and Analysis. *Procedia Computer Science*, *230*, 193–202. https://doi.org/10.1016/j.procs.2023.12.074

Perini, A. (2019). Data-Driven Requirements Engineering. The Supersede Way. *Communications in Computer and Information Science*, 13–18. https://doi.org/10.1007/978-3-030-11680-4_3

Rahali, A., & Akhloufi, M. A. (2023). End-to-End Transformer-Based Models in Textual-Based NLP. *AI*, *4*(1), 54–110. https://doi.org/10.3390/ai4010004

Vogelsang, A., & Borg, M. (2019). Requirements Engineering for Machine Learning: Perspectives from Data Scientists. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 245–251. https://doi.org/10.1109/rew.2019.00050