Aspect-based Sentiment Analysis with an Ensemble Learning Framework for Requirements Elicitation from App Reviews

Zhiquan An[1], Teng Xiong[1], Zhiyuan Zou[1], Hongyan Wan[1,2]*

[1] *School of Computer Science and Artificial Intelligence, Wuhan Textile University, China*
[2] *Engineering Research Center of Hubei Province for Clothing Information, Wuhan Textile University, China*
1499231140@qq.com, 1027647011@qq.com, 249948870@qq.com, why0511@whu.edu.cn

Abstract

In the past three years, numerous studies have demonstrated the effective performance of aspect-based sentiment analysis (ABSA) in eliciting requirements from APP reviews. However, in aspect category detection (ACD) and aspect category polarity (ACP), traditional supervised machine learning techniques still dominate the latest research. Inherent limitations, such as poor generalization ability, low robustness, and high dependence on feature engineering, often constrain the methods. Additionally, while most research continues to center on the efficacy of singular models or strategies, the effective amalgamation of traditional and contemporary techniques has yet to be adequately explored. Given these challenges, this study proposes an ensemble learning framework based on XGBoost and Stacking. Compared to baseline, the framework achieves a performance improvement of 22.9%-28.4% in ACD and 9.3%-13.2% in ACP tasks based on different feature engineering. Overall, the preliminary attempt in this study to apply ensemble learning in ABSA for requirements elicitation from APP reviews, providing a feasible new direction for overcoming the inherent limitations of traditional techniques in fine-grained sentiment modeling in this field.

Keywords: APP reviews, Aspect-based sentiment analysis, Ensemble learning, Requirements elicitation

1 Introduction

An increasing number of users share their feedback, recommendations, and suggestions on software products through reviews and ratings in the APP stores [1]. These feedbacks shape the potential first impression for other users. For developers, APP reviews reflect the product acceptance level and market position, and provide an important avenue to capture core needs, enhance user experience, and maintain market competitiveness. In the academic community, automated analysis of APP reviews has gradually become a research hotspot in the requirements elicitation field [2].

Aspect-Based sentiment analysis (ABSA) [3] is a critical natural language processing (NLP) technology. While traditional sentiment analysis (SA) typically centers on recognizing a single topic, ABSA is crafted to address reviews that touch on multiple aspects, each possibly carrying a unique sentiment polarity. In recent years, ABSA has demonstrated significant effectiveness in the APP reviews-based requirements elicitation field, and its research popularity continues to grow. As illustrated in Figure 1, ABSA primarily consists of two critical tasks: Aspect-Category Detection (ACD) and Aspect-Category Polarity (ACP). ACD aims to categorize user reviews into predefined categories, identifying the main topics users are addressing. ACP focuses on assigning sentiment values to each identified category, indicating whether the expressed sentiment towards a particular feature is positive or negative.

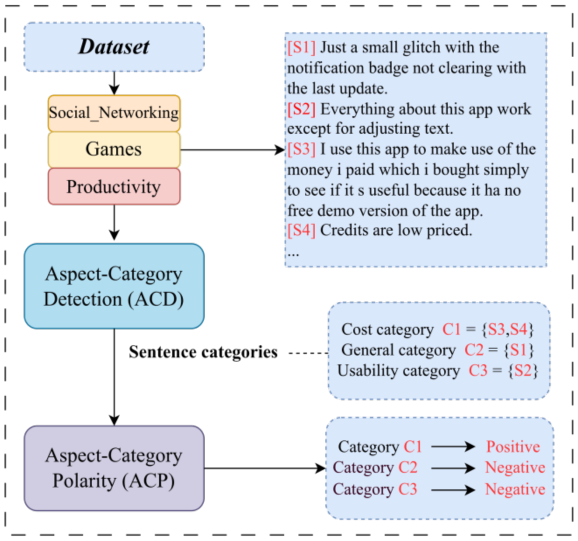

Dataset
Social_Networking
Games
Productivity
[S1] Just a small glitch with the notification badge not clearing with the last update.
[S2] Everything about this app work except for adjusting text.
[S3] I use this app to make use of the money i paid which i bought simply to see if it s useful because it ha no free demo version of the app.
[S4] Credits are low priced.
...
Aspect-Category Detection (ACD)
Sentence categories
Cost category C1 = {S3,S4}
General category C2 = {S1}
Usability category C3 = {S2}
Aspect-Category Polarity (ACP)
Category C1 → Positive
Category C2 → Negative
Category C3 → Negative

Figure 1. The ACD and ACP tasks in aspect-based sentiment analysis

Most of the research on ACD and ACP still relies on traditional supervised machine learning (ML) methods in the field of ABSA for requirements elicitation from App reviews. It is still focused on the effectiveness or empirical evaluation of a single model or strategy. Inherent limitations, such as lack of generalization ability, low robustness, and high dependence on feature engineering often constrain these methods. In addition, there are still significant research gaps in exploring how to efficiently integrate sentiment lexicons, traditional machine learning, or modern deep learning methods to deal with complex textual data.

To alleviate these problems, this study constructs a

*Corresponding Author: Hongyan Wan; E-mail: why0511@whu.edu.cn
DOI: https://doi.org/10.70003/160792642024122507012

stacking-based [4-5] ensemble learning framework [6]. We designed a series of cross-validation experiments and eventually selected logistic regression (LR), support vector machine (SVM), and multilayer perceptron (MLP) as base models, while introducing random forest (RF) to increase robustness and reduce overfitting, as well as XGBoost [7] to improve stability. The overall meta-model adopts LR. Compared with the baseline and benchmark, this architecture alleviates potential shortcomings of individual models, enhances model robustness, and effectively improves the generalization ability to handle different tasks. The main contributions are as follows:

1) Compared with the baseline, the proposed ensemble framework demonstrates significant performance improvement and high generalization ability in ACD and ACP tasks, surpassing the benchmark in specific domains. It provides a new perspective for further exploration and improvement of integrated models in the ABSA field.

2) A design empirical study is conducted to explore the performance impact of different feature engineering on the stacking-based ensemble. Revealing the tremendous potential of fine-tuning in capturing complex semantic relationships and adapting to specific tasks.

3) The impact of the interaction between the base model and the meta-model of the integrated framework on ABSA performance is explored and revealed through cross-validation experiments.

The remainder of the paper is organized as follows: Section 2 introduces the related works. Section 3 presents the approach of this study. Section 4 proposes the results of question-oriented experiments. Section 5 discusses the threats to the validity. Section 6 summarizes our study and provides future works.

2 Related Work

This section summarizes the application of SA in the field of requirements elicitation from APP reviews.

2.1 Traditional Sentiment Analysis Methods

Traditional SA uses topic models (such as Latent Dirichlet Allocation) to detect the topics in reviews and creates topic-based sentiment models using sentiment lexicons (such as Sentistrength and Senti4SD), ML, and deep learning sentiment classifiers. Al-Hawari et al. [11] introduced an ACRM method to automatically categorize requirements derived from app feedback. Gao et al. [12] introduced PRISharer, a framework for extracting potential licensing requirements from app feedback. Iqbal et al. [13] designed a framework that combines deep learning and ensemble learning to perform sentiment analysis on Pashto text in social media. Using ML and a sentiment lexicon, Jha and Mahmoud [14] addressed the challenge of identifying and categorizing non-functional requirements (NFRs) within user feedback. Fu et al. [15] unveiled the WisCom system, employing an SA technique anchored on the LDA topic model, to discern user needs from tri-level ratings and critiques on app platforms.

2.2 Aspect-based Sentiment Analysis Methods

The ABSA method uses ML, deep learning, and other methods to extract feature words representing certain aspects in sentences and analyze the corresponding sentiments. In the field of requirements elicitation from app reviews, the ABSA has relatively few applications. Araujo et al. [8] classified and empirically studied word embedding techniques, and demonstrated that the BERT families performed well in all prediction tasks, and was more suitable for handling semantic similarity in evaluating texts. Alturaief et al. [9] introduced the only publicly available dataset in this domain named AWARE, which comes with detailed aspect terms and sentiment annotations, and proposed a baseline. After this dataset, they conducted an empirical study and found that the ML models trained with LR and BERT embeddings outperformed other models in terms of performance [10].

The current research mainly focuses on the effectiveness evaluation of individual models or strategies. This paper introduces an ensemble learning framework based on the stacking architecture, which seamlessly integrates linear, tree, and base models.

3 Methodology

3.1 Overview

The methodology is divided into five parts: preprocessing, feature engineering, aspect category detection (ACD) task, and aspect category polarity (ACP) task. The overall framework is shown in Figure 2. To ensure the reproducibility of the experiments, we have open-sourced both the code samples and the dataset[1].

3.2 Preprocessing

The dataset is preprocessed by the NLTK tools [16]. The process included tokenization, removing stop words, eliminating non-alphabetic characters, and part-of-speech tagging (POS) [17]. In the stop word removal task, we add some commonly used words specific to app reviews, such as ‘app’, ‘download’, and ‘update’, to ensure that these words do not affect the quality of the features. In the POS task, in addition to nouns (NN) and adjectives (JJ), verbs (VB) are additionally retained, as our experiments have found that retaining VB can improve the performance of the base model in ensemble learning.

3.3 Feature Engineering

Both frequency-based and neural language model-based representations are used to verify the generalization performance of the proposed ensemble learning framework.

3.3.1 Frequency-based Representation

In frequency-based representation, the words serve as its feature value. The term frequency-inverse document frequency (TF-IDF) [18] is adopted to extract text features. When applying the representation method, we select appropriate parameters and thresholds to ensure the maximum capture of crucial information in APP reviews. This provides helpful features for improving performance in subsequent

1 https://github.com/Zhiquan-An/ABSA-for-app-reviews-based-requirements-elicitation

models. Additionally, a customized IDF weighting method is introduced to reduce the weight of words that appear in over 50% of the reviews. This helps to reduce the weight of some overly common words that are not significant for SA.

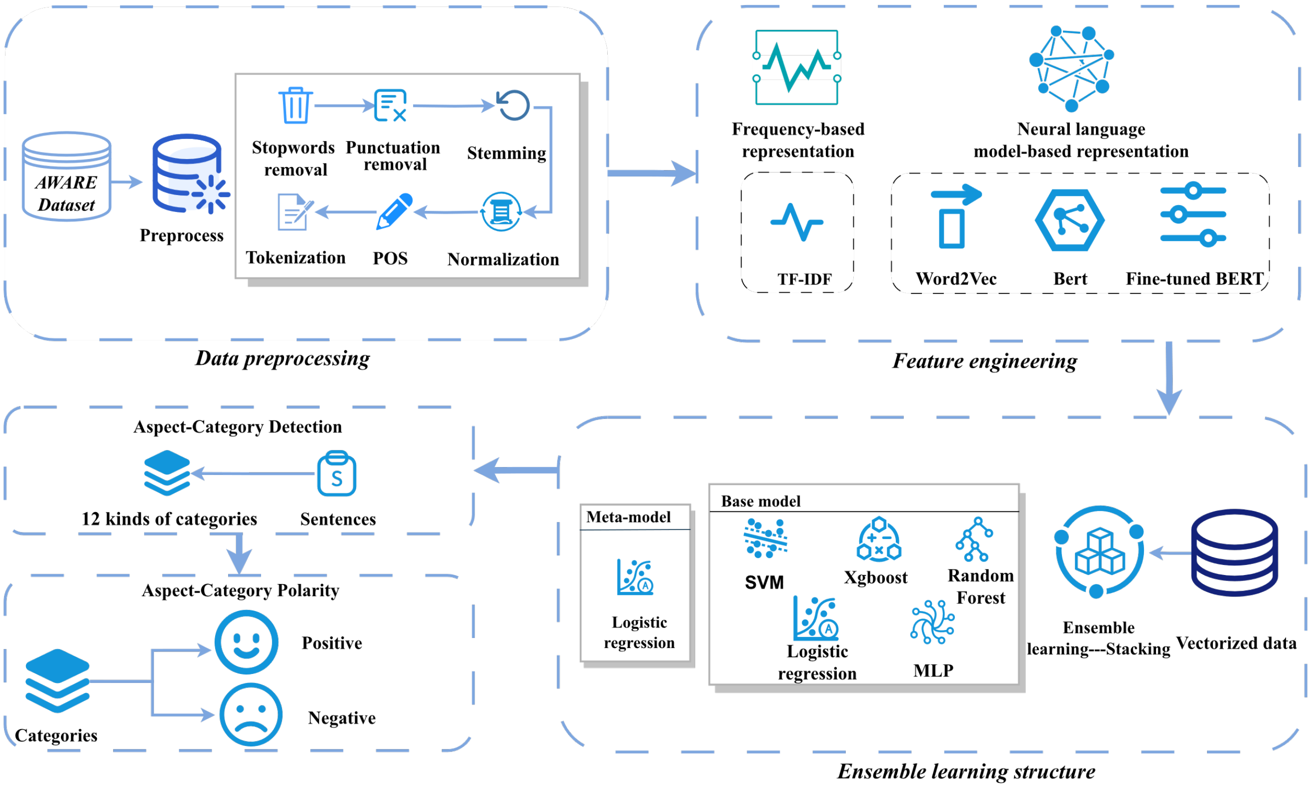


Figure 2. The Overview of the ensemble framework

3.3.2 Neural Language Model-based Representation

In neural language model-based representation, pre-trained word embedding models are used: Word2vec [19], BERT [20], and fine-tuned BERT.

Word2vec using the Gensim library and chose a 300-dimensional vector space. Normalized word embeddings and addressed out-of-vocabulary (OOV) issues in the reviews to ensure the robustness of the model.

Using HuggingFace[2] Transformers library, we select the base model of BERT ('bert-base-uncased') and take the hidden state of the first token outputted by BERT (which typically represents the meaning of the entire sentence) as the representation of the entire review. This representation of each review is then inputted into the ensemble model.

In order to ensure that BERT provides the best [CLS] representation for downstream tasks on our dataset, we have fine-tuned the 'bert-base-uncased' version. The entire dataset is divided into 85% training data and 15% validation data. In the experiment, we follow the advice of the original authors of BERT and select the following hyperparameter configuration for multiple rounds of testing: batch Size: 12, 14, 16, 32, learning rate (Adam): 3e−5, 2e−5,4e−5, 5e−5, number of epochs: 2, 3, 4, 5. After multiple rounds of experiments, the hyperparameter combination that yielded the best performance on the validation data is selected to ensure the model's performance and robustness in practical applications.

2 https://huggingface.co/

3.4 The Ensemble Learning Framework

This study chooses an ensemble learning model based on the Stacking architecture (see Figure 2) and XGBoost. In order to better capture different patterns and dependencies in the reviews, the base models selected include linear model 'LR', non-linear model 'multi-layer perceptron (MLP)', tree-based models 'random forest (RF)' and 'XGBoost', as well as 'support vector machines (SVM)' based on maximum margin. The outputs of the base models are provided as new features to the meta-model, with LR selected as the meta-model to balance the outputs of the base models and achieve a comprehensive prediction. Due to its ability to linearly combine and regularize in the ensemble, overfitting is avoided.

In addition, a weighting mechanism is introduced for the predictions of the base model, where the weights are determined based on the F1 scores of each model on the validation set. The base model's performance on the validation set reflects the reliability of its predictions, so a base model with better performance should have a more significant impact on the input of the meta model. The calculation method for the weights is as follows:

$$Meta\_X[i] = \sum_{j=1}^{N} F1_j \times pred\_proba_j[i] \qquad \textbf{(1)}$$

For the given sample i:

j is the index of the base model, ranging from 1 to N

(where N is the total number of base models)

$pred\_proba_j$ denotes the probability that the base model indexed by j predicts the positive class for the sample.

$F1_j$ is the F1 score of the base model indexed by j on the validation set.

Table 1 provides the pseudocode for the methodology. Lines 1-3 initiate five base models and decompose the dataset 'D' into training and test subsets. Lines 4-6 train each base model M on the training set. Lines 7-12 involve predicting with each base model M on the validation set V, computing the respective F1 scores, and storing them. Lines 14-17 are dedicated to calculating the weights. Finally, lines 18-20 train the meta-model, Logistic Regression, on the StackingInput and the labels from the validation set, producing the ultimate prediction P.

Table 1. Ensemble algorithm

```
Input: Training Dataset D, Validation set V
Output: Final Prediction P
1. Function EnsembleLearningModel(D, V)
Split D into training and test sets as 10-flod
3.  Initialize base models: LR, MLP, XGBoost, RF, SVM
4.  For each base model M in [LR, MLP, XGBoost, RF, SVM]:
5.      Train M on training set
6.  End for
7.  Initialize an empty dictionary Predictions{}
8.  Initialize an empty dictionary F1Scores{}
9.  For each base model M in [LR, MLP, XGBoost, RF, SVM]:
10.     Predictions[M] = M.predict(V)
11.     F1Scores[M] = ComputeF1Score(Predictions[M], V.labels)
12. End for
13. TotalScore = sum(F1Scores.values())
14. Weights = {}
15. For each model M in F1Scores:
16.     Weights[M] = F1Scores[M] / TotalScore
17. End for
18. StackingInput = Concatenate predictions using Weights
19. Train meta-model LR on StackingInput and V.labels
20. P = meta-model LR.predict(test set)
21. Return P
```

Overall, our ensemble method considers both linear and non-linear models, as well as various model characteristics, ensuring that the model has high adaptability and robustness.

3.5 Aspect Category Detection (ACD)

ACD task is commonly referred to as identifying and extracting aspect keywords in sentences. As shown in Figure 2, this study uses a constructed ensemble learning classifier to perform aspect extraction on reviews of three categories of apps (Productivity, Social networking, and Games). There are a total of 10 aspect categories according to the predefined aspect lexicon [9]. In addition, comparative experiments are conducted to comprehensively evaluate the impact of different feature engineering strategies and reveal the optimal aspect extraction strategy.

3.6 Aspect Category Polarity (ACP)

ACP focuses on assigning a sentiment polarity to each extracted aspect term. The constructed ensemble model performs ABSA under different feature engineering settings. Considering the potential data imbalance issue, SMOTE is employed for oversampling to ensure sufficient training for each sentiment label. Additionally, ten-fold cross-validation is applied to both the base models and the meta-model to ensure model generalization and reduce possible biases.

4 Experiments and Results Analysis

4.1 Research Questions

The main objective of this study is to identify the effectiveness of our ensemble learning framework in ACD and ACP tasks under different feature engineering backgrounds in ABSA for APP reviews-based requirements elicitation. To guide the experiments, the following research questions are proposed:

RQ1: In the ACD and ACP tasks, has the performance of the ensemble learning framework improved?

RQ2: What changes in the performance of the ensemble learning framework occur when applying different feature engineering techniques?

RQ3: How do the interactions between base models and the meta-model affect the SA performance?

4.2 Dataset

Our previous systematic mapping study (Accepted at APSEC 2023 conference) has systematically evaluated the data resources in this field. AWARE [9] is currently the only recognized and publicly available dataset in the field [10]. Therefore, it is selected as the benchmark dataset for this study. AWARE consists of 11,323 app reviews from three domain apps (Productivity, Social networking, Games), annotated with aspect terms, categories, and sentiment.

4.3 Metrics

In order to objectively evaluate the performance of our model on aspect classification and sentiment polarity prediction tasks, this study adopts the following commonly used evaluation metrics:

(1) F1-Score: It is the harmonic mean of precision and recall. It balances the consideration of false positives and false negatives, providing an overall assessment of the model's performance.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision \pm Recall} \quad \textbf{(2)}$$

(2) Accuracy: The proportion of samples predicted to be positive that are actually positive.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(3)}$$

4.4 Results and Analysis

4.4.1 Results and Analysis for RQ1

In order to verify whether the performance of the ensemble model framework has been improved, the baseline [9] and the benchmark based on ML and Deep Learning (BERT) [10] are selected for comparative experiments.

In the ACD task (Figure 3), our ensemble framework enhances the f1-score for social networking, games, and productivity app reviews by 22.9%, 28.4%, and 24.4% over the baseline. Compared with the benchmark, social networking decreased by 0.8%, while games and productivity increased by 2.9% and 0.4%. This indicates that the ensemble framework is significantly superior to the baseline in all scenarios and demonstrates universality and robustness. Although we slightly lag behind the benchmark in social networking, the leading position in games and productivity proves the competitiveness and advantages of our method.

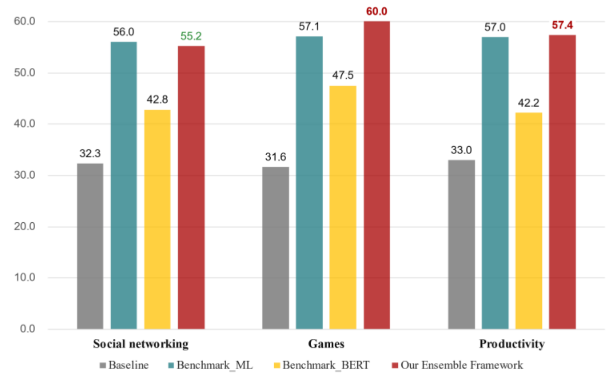


Figure 3. Performance of the ensemble framework in ACD tasks (f1-score)

In the ACP task (Figure 4), compared to the baseline, the accuracy of reviews on social networking, games, and productivity app reviews increases by 11.5%, 9.3%, and 13.2%. Compared to the benchmark, there is a 2.4% improvement in productivity and a slight decrease of 0.6% and 0.9% in social networking and games. This indicates that our framework has significantly improved compared to the baseline in all scenarios, especially in productivity application reviews. However, the results for social networks and games are inadequate compared to the benchmark, and further optimization may be needed for these domains.

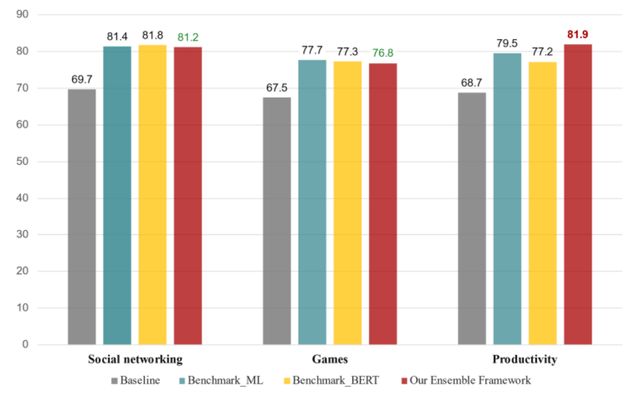


Figure 4. Performance of the ensemble framework in ACP tasks (accuracy)

From the results of ACD and ACP, the proposed integrated framework has demonstrated its generalization performance and effectiveness in multiple dimensions and scenarios. However, reviews in specific scenarios may contain unique text characteristics or challenges, resulting in a slightly inferior benchmark. Nevertheless, these discrepancies highlight new directions and challenges for future research, inspiring us to delve further into and optimize various aspects of the model.

4.4.2 Results and Analysis for RQ2

Figure 5 and Figure 6 show the performance trends of the ensemble framework in the ACD and ACP tasks using TF-IDF, Word2vec, BERT, and fine-tuned BERT.

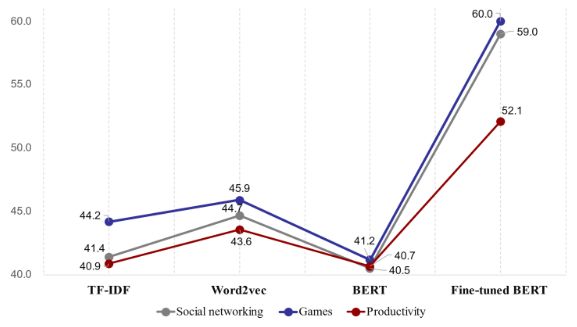


Figure 5. Feature engineering performance in ACD tasks

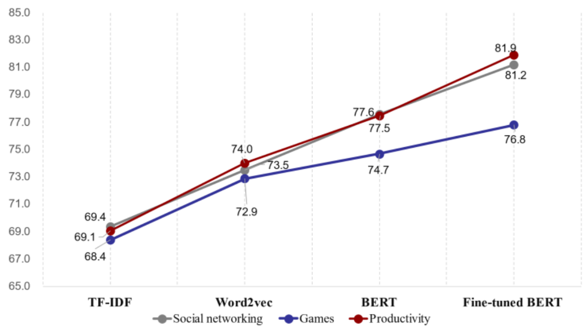


Figure 6. Feature engineering performance in ACP tasks

In the ACP tasks, the performance of fine-tuned BERT is the best, and Word2vec outperforms TF-IDF. The f1-score of the original BERT performs relatively weaker than others because BERT is initially designed to capture deep semantic relationships in long texts. At the same time, the ACD tasks may rely more on local semantics and word frequency recognition. BERT subword tokenization mechanism may divide key aspect words into smaller parts, affecting the complete recognition of words. The performance of fine-tuned BERT significantly improves, mainly because the fine-tuning process precisely adjusts the weights of BERT and further explores specific semantic structures at the aspect level, making it more suitable for the specific requirements of the ACD task. The productivity category performs relatively less than others in the fine-tuned BERT model. This could be attributed to the fact that reviews for productivity applications often delve into more technical details and use specialized terminology. Therefore, accurately modeling the complexity and details of this specific domain remains a challenge.

In the ACP tasks, various word embedding techniques have shown a performance increment trend from basic to advanced. As a traditional method, TF-IDF has the lowest accuracy in all categories, while fine-tuned BERT performs best. Compared to TF-IDF and Word2vec, the original BERT has already demonstrated strong performance. However, its capability is further amplified after fine-tuning, where the improvement is most significant. This highlights the crucial role of fine-grained, task-specific fine-tuning in optimizing model performance and the potential to capture complex semantic relationships.

4.4.3 Results and Analysis for RQ3

In order to gain a deeper understanding of how the interaction between base models and meta models affects the SA performance in the ensemble framework, a series of cross-validation experiments are designed. To ensure consistency of variables, the original BERT is used as the feature engineering, and all models are trained using ten-fold cross-validation to obtain the average f1-score and accuracy.

Table 2 shows the results of SA using individual base models. The f1-score rankings are highlighted in bold. Among all APP categories, SVM and LR demonstrate strong performance, especially SVM leading in the 'Social Networking' and 'Productivity' categories, while LR is slightly ahead in the 'Games' category. RF performs relatively weakly, particularly in the 'Games' category. XGBoost and MLP show slightly fluctuating performance across different APP categories, but maintain a certain level of robustness overall. When selecting models for different APP categories, considering specific data characteristics and the inherent advantages of the models is particularly important to ensure optimal SA results.

Table 2. Performance for a single base model (f1-score)

APP Categories	Base models				
	XGBoost	LR	MLP	RF	SVM
Social networking	**74.36**	**76.61**	73.85	71.46	**77.12**
Games	**73.31**	**74.34**	71.70	69.51	**74.57**
Productivity	74.10	**77.40**	**75.60**	72.50	**76.90**

Considering the standard practices and guiding principles of the Stacking framework [5, 21] (number of base models > 3), we choose to conduct comparative experiments using the top three and four performing base models. The results are shown in Table 3.

By comparing different base model combinations, it is evident that the choice of base model combination significantly influences the final performance. For instance, in the social networking category, adding MLP to the 'SVM+LR+XGBoost' combination improves the performance from 75.3 to 76.1. In contrast, this combination enhances the result even more significantly in the Games category, from 72.7 to 73.8. Although in specific scenarios adding more base models might yield performance improvements, blending a single model or a limited combination might lead to performance stagnation or even decline, as evidenced by the mere 0.2 improvement in the Productivity category upon adding XGBoost to the 'SVM+LR+MLP' combination. Our framework integrates relationship weights among various models, achieving optimal ensemble results and demonstrating outstanding and consistent performance across all APP categories.

Table 3. Performance for a base model combination (accuracy)

Base model combination	APP Categories		
	Social networking	Games	Productivity
SVM+LR+XGBoost	75.31	71.88	75.65
SVM+LR+MLP	74.81	71.66	76.18
SVM+LR+XGBoost+MLP	76.07	73.60	76.33
SVM+LR+XGBoost+RF	75.93	73.1	77.21
Our ensemble frame work	**77.60**	**74.70**	**77.50**

Table 4. The influence of meta models in the ensemble framework (accuracy)

Different meta models	APP Categories		
	Social networking	Games	Productivity
SVM	76.87	74.17	77.14
MLP	76.80	73.88	75.09
RF	68.09	65.72	68.92
XGBoost	99.89	99.81	99.71
XGBoost+early stopping	77.3	74.10	78.22
LR (our strategy)	**77.60**	**74.70**	**77.50**

Table 4 illustrates the influence of different meta-model choices on the performance of the ensemble learning framework. Among all APP categories, SVM as the meta-model demonstrates relatively robust performance, especially in the 'Productivity' category, with an accuracy of up to 77.9%. This may be because SVM can find a better decision boundary when dealing with the output predictions of multiple base models. MLP performs slightly worse than SVM. It is worth noting that RF demonstrates significantly lower performance across all categories. Compared to a single classifier, the meta-classifier based on RF diminishes the overall performance. This may be attributed to the unidirectional integration of multiple decision trees, resulting in model redundancy and excessive averaging, reducing performance.

In addition, XGBoost displays tendencies of overfitting. This might be attributed to the limited training data at the meta-model layer of Stacking, and XGBoost, a highly optimized algorithm, tends to overlearn. Simultaneously, without parameter adjustments, the sensitivity of XGBoost to slight data variations might be magnified at the meta-model level. As a result, an early stopping mechanism is introduced to counteract this overfitting. In specific categories, LR demonstrates performance akin to, or even slightly surpassing, SVM. As a meta-model, LR maintains a commendable balance and is less susceptible to overfitting.

Overall, choosing the appropriate base models and meta-models is crucial for performance in the ensemble learning framework. Individual base models like SVM and LR have demonstrated exceptional capabilities in multiple application scenarios. However, the real essence lies in how to combine them properly. There are better strategies than stacking models; strategic considerations are required when balancing and coordinating various models. Regarding meta-model selection, although SVM is competitive in some scenarios, LR has secured a dominant position due to its robust and consistent performance. In conclusion, strategically combining base models and selecting LR as the meta-model is our method to achieve optimal performance.

5 Validity Threats

This section focuses on potential threats and how to control or alleviate them. The four validity threats are as follows:

Construct validity: A stacking approach with machine learning base models and meta-models has been employed, which might potentially influence the reliability of the outcomes. To address this concern, cross-validation experiments (section 4.4.3) have been conducted to validate the reliability of the model selection.

External validity: The experiment uses AWARE datasets, which is publicly available, commonly used, and authentic. While concerns might arise regarding the generalizability due to reliance on a single dataset, it is imperative to note that AWARE represents the only well-recognized and established dataset in this particular research domain. This can reduce the threat of this study producing different results in different datasets. Also, we will introduce new datasets in the future.

Internal validity: The experimental results indicate that ensemble learning is very suitable for ABSA. However, the selection of different base models and meta-models can result in significant differences in experimental results.

Conclusion validity: A detailed research design is designed for this study. Section 4.1 shows the research question, and the experimental results, provided a detailed analysis of the questions to ensure the conclusions are valid.

6 Conclusion and Future Work

This study explores the application of the proposed stacking-based ensemble framework in ABSA for requirements elicitation from app reviews. Compared to the baseline, our integrated learning framework has improved the F1 score by 22.9% to 28.4% in the ACD task, and the accuracy has increased by 9.3% to 13.2% in the ACP task. At the same time, both tasks demonstrate robust performance compared to the baseline, even surpassing it in certain categories. These results fully demonstrate the effectiveness and generalization performance of our framework.

Empirical research has shown that in the ensemble learning framework based on the Stacking architecture, the fine-tuned BERT outperforms other feature engineering methods, such as TF-IDF, Word2vec, and the original BERT, when dealing with technical reviews. This underscores the immense potential of fine-tuning within the ensemble learning framework.

Through cross-validation, we assessed how base and meta-model choices affect the ensemble framework. It is not about simply stacking such as the RF meta-model, which can diminish performance. The choice of individual models and their strategic integration are critical. Based on F1-weighting, we adopted five models as base models and LR as the meta-model. This method ensures our framework's efficiency and robustness.

The study needs further improvement. The next step is to exploring the impact of deep learning on the ensemble framework and conducting comparative experiments with other methods.

Acknowledgment

This work is supported by the National Natural Science Foundation of China Project (No. 62102291), and the Opening Foundation of Engineering Research Center of Hubei Province for Clothing Information (No. 2022HBCI02, No. 2022HBCI05).

References

[1] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, H. C. Gall, How can I improve my app? Classifying user reviews for software maintenance and evolution, *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Bremen, Germany, 2015, pp. 281–290.

[2] N. Genc-Nayebi, A. Abran, A Systematic Literature Review: Opinion Mining Studies from Mobile App Store User Reviews, *Journal of Systems and Software*, Vol. 125, pp. 207–219, March, 2017.

[3] G. Brauwers, F. Frasincar, A Survey on Aspect-Based Sentiment Classification, *ACM Computing Surveys*, Vol. 55, No. 4, Article No. 65, April, 2023.

[4] D. H. Wolpert, Stacked Generalization, *Neural Networks*, Vol. 5, No. 2, pp. 241–259, January, 1992.

[5] M. H. D. M. Ribeiro, L. dos Santos Coelho, Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series, *Applied soft computing*, Vol. 86, Article No. 105837, 2020.

[6] A. Yusup, D. Chen, Y. Ge, H. Mao, N. Wang, Resource Construction and Ensemble Learning based Sentiment Analysis for the Low-resource Language Uyghur, *Journal of Internet Technology*, Vol. 24, No. 4, pp. 1009–1016, July, 2023.

[7] K. Afifah, I. N. Yulita, I. Sarathan, Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier, *2021 International Conference on Artificial Intelligence and Big Data Analytics*, Bandung, Indonesia, 2021, pp. 22–27.

[8] A. F. Araujo, M. P. S. Gôlo, R. M. Marcacini, Opinion mining for app reviews: an analysis of textual representation and predictive models, *Automated Software Engineering*, Vol. 29, No. 1, pp. 1–30, May, 2022.

[9] N. Alturaief, H. Aljamaan, 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)*, Melbourne, Australia, 2021, pp. 211–218.

[10] N. Alturayeif, H. Aljamaan, J. Hassine, An automated approach to aspect-based sentiment analysis of apps reviews using machine and deep learning, *Automated Software Engineering*, Vol. 30, No. 2, Article No. 30, November, 2023.

[11] A. Al-Hawari, H. Najadat, R. Shatnawi, Classification of application reviews into software maintenance tasks using data mining techniques, *Software Quality Journal*,

Vol. 29, No. 3, pp. 667–703, September, 2021.

[12] H. Gao, C. Guo, G. Bai, D. Huang, Z. He, Y. Wu, J. Xu, Sharing runtime permission issues for developers based on similar-app review mining, *Journal of Systems and Software*, Vol. 184, Article No. 111118, February, 2022.

[13] S. Iqbal, F. Khan, H. U. Khan, T. Iqbal, J. H. Shah, Sentiment Analysis of Social Media Content in Pashto Language using Deep Learning Algorithms, *Journal of Internet Technology*, Vol. 23, No. 7, pp. 1669–1677, December, 2022.

[14] N. Jha, A. Mahmoud, Mining non-functional requirements from App store reviews, *Empirical Software Engineering*, Vol. 24, No. 6, 3659–3695, December, 2019.

[15] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, 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 (KDD)*, Chicago, Illinois, USA, 2013, pp. 1276–1284.

[16] S. Bird, E. Klein, E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*, O'Reilly Media, Inc., 2009.

[17] M. Banko, R. C. Moore, Part-of-speech tagging in context, *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 2004, pp. 556–561.

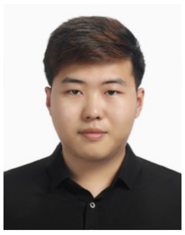
[18] J. Ramos, Using tf-idf to determine word relevance in document queries, *Proceedings of the first instructional conference on machine learning*, Vol. 242, No. 1, pp. 29–48, December, 2003.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, 2013, Lake Tahoe, Nevada, USA, pp. 3111–3119.

[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*, October, 2018. https://arxiv.org/abs/1810.04805

[21] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, P. N. Suganthan, Ensemble deep learning: A review, *Engineering Applications of Artificial Intelligence*, Vol. 115, Article No. 105151, October, 2022.

Biographies



Zhiquan An works as an assistant engineer at Dongfang Electronics Co., Ltd. In September 2022, he joined PhD. Bangchao Wang's research group and became a master student of Wuhan Textile University in China. His research interests include requirements engineering, sentiment analysis, natural language processing, etc.

Teng Xiong received the B.S. degrees in Digital Media Technology from Hunan University of Science and Engineering in 2022. In September 2023, he joined Prof. Xinrong Hu's research group became a master student of Wuhan Textile University in China. His research interests include requirement engineering, natural language processing, etc.

Zhiyuan Zou joined Prof. Bangchao Wang's research group became a master student of Wuhan Textile University in April 2023. His research interests include software traceability, software engineering and natural language processing, etc.

Hongyan Wan received the Ph.D. degree from Wuhan University in 2021. She is currently a lecturer in the School of Computer Science and Artificial Intelligence at Wuhan Textile University. Her research interests include software engineering, natural language processing, machine learning, and intelligent algorithms.