Research On Software Testing Method Based on Social Media Risk Control

Xiaofan Zhang

Xi'an Eurasian University, Xi'an, Shaanxi, China

e-mail address:zhangxiaofan@eurasia.edu

Abstract. With the popularization of social media, the management and risk control of user comments have gradually become an important issue. In order to improve the effectiveness of the online comment risk control system, this paper proposes a software test-based method to evaluate the execution effect of the existing risk control strategy by releasing test comment data. The test method designs a variety of scenarios to simulate different types of risky comments and analyzes the expected disposition results in comparison with the actual processing results. Through this method, we are able to quantify the effectiveness of the risk control strategy and identify potential deficiencies, thus providing a basis for subsequent strategy optimization. The experimental results show that this method not only improves the accuracy of risk control, but also provides a feasible path to realize automated monitoring.

Keywords: Risk Control; Testing Methodology; Automated Monitoring.

1 Introduction

In the era of rapid development of information dissemination, social media platforms carry a large number of user interactions and comments. However, the ensuing negative information, false statements and inappropriate comments pose serious challenges to social opinion and the online environment[1]. As a result, many social media platforms have established comment risk control systems to ensure the health and safety of comment content. However, the effectiveness of risk control strategies is often difficult to quantify and evaluate, which makes the improvement of these systems face many difficulties.

To address this problem, this paper proposes an innovative testing methodology that aims to verify the effectiveness of existing risk control strategies through real-world operations. This study employs simulated test comments to construct several test scenarios covering various possible types of risky comments, such as malicious attacks, rumor spreading, and harassing messages. We compare the expected disposition results with the results of the actual system response, and then evaluate the accuracy and timeliness of the risk control strategy. This method not only provides a sci-

Z. A. Zukarnain et al. (eds.), *Proceedings of the 2024 International Conference on Artificial Intelligence and Digital Management (ICAIDM 2024)*, Advances in Intelligent Systems Research 187,
https://doi.org/10.2991/978-94-6463-578-2_24

entific basis for the evaluation of the effectiveness of risk control, but also lays the foundation for future strategy adjustment and optimization.

The structure of this paper is as follows: the second part introduces the relevant theories and existing research, the third part elaborates the design and implementation of the test method, the fourth part shows the experimental results and discusses them, and the fifth part summarizes the research results and proposes future research directions. Through this research, we hope to provide new ideas and methodological support for the development of social media comment wind control system.

2 The Relevant Theories and Existing Research

2.1 Theoretical Basis of Risk Control Strategy

Risk control strategy is mainly based on the theoretical framework of risk management, aiming at identifying, assessing and controlling potential risks. In the social media environment, the risk of comment content mainly manifests itself in the form of negative impact on social opinion, legal liability and brand image damage. Therefore, the risk control strategy should cover the following aspects:

Risk identification: identify potential risky comments, including malicious remarks, false information, etc., through natural language processing (NLP) technology. Commonly used methods include sentiment analysis, keyword extraction and topic modeling[2].

Risk Control: Formulate appropriate risk control measures based on the assessment results, such as deleting comments, prohibiting users from speaking, and increasing the frequency of reviewing comments.

2.2 Research Status of Existing Risk Control Systems

In recent years, the research on social media risk control system has made some progress. The existing research mainly focuses on the following directions:

Algorithm optimization: many researches are devoted to improving the machine learning and deep learning algorithms used in the wind control system. For example, Convolutional Neural Networks (CNN)[3]and Recurrent Neural Networks (RNN)[4]are widely used in text categorization tasks to improve the detection accuracy of risky comments.

User behavior analysis: more and more studies focus on the relationship between user behavior and comment risk. Potential malicious users can be identified more accurately by analyzing users' historical comment records, interaction behaviors, and social network structure[5].

Multi-modal data fusion: with the increasing richness of social media content, researchers have begun to explore how to combine multiple data types, such as text, images, and videos, in order to improve the comprehensive performance of the risk control system[6].

2.3 Research Status of Testing Methods

Although there have been a number of studies on wind control strategies, testing methods for wind control systems are still weak. The current research mainly focuses on the following aspects:

A/B testing: some platforms adopt the A/B testing method to compare the effect of different risk control strategies by running them simultaneously. This method is intuitive and easy to implement, but may be affected by sample selection bias.

Performance evaluation metrics: Performance evaluation metrics: existing research mostly focuses on traditional performance metrics such as precision rate and recall rate, but in the field of wind control these metrics often fail to fully reflect the effectiveness of the system. Therefore, there is an urgent need to develop new evaluation metrics to better reflect the effectiveness of risk control strategies.

3 Relevant Theories and Existing Research

The overall design thinking can be elaborated in the following key steps.

3.1 Design Thinking

In this section, it is first necessary to define the purpose and scope of the test. The main goal of testing is to verify the effectiveness of the online policy rules and whether the generated use cases accurately reflect the expected results.

3.2 Data Reading

Writing code to read the online database: a script is written using an appropriate programming language to connect to the online database and extract the policy rules.

data parsing: structure the acquired data by parsing the drools rules for subsequent use case assembly.

3.3 Use Case Assembly

User Information Matching: Based on the information of the comment publisher, find the eligible users in the account pool. This step can use database query statements or other data processing techniques.

Comment Content Generation: Automatically generate the corresponding comment content according to the requirements of the policy rules.

Exemption rule judgment: according to the pre-set exemption rules, the found users and comment contents are judged to ensure that they meet the requirements.

Disposition setting: according to the rule definition, determine the corresponding disposition, such as whether it is necessary to mark, delete or notify.

3.4 Use Case Validation

Actual state validation: put the assembled use cases into the actual environment to run and record the actual results.

Result Comparison: Compare the actual results with the expected results to determine whether the use case passes or fails.

Consistency: If the actual result is consistent with the expectation, it is recorded as a success case.

Inconsistency: If the actual result is inconsistent with the expectation, the failure use case needs to be recorded and relevant information should be collected for analysis.

3.5 Notification Mechanism

Email notification mechanism: When the use case validation fails, the email notification mechanism is immediately activated. The email should contain detailed information about the failed use case, such as the use case number, the difference between the actual result and the expected one, and the time of occurrence. At the same time, it should also provide the relevant cid so that developers and testers can quickly locate the problem.

Issue tracking and resolution: After receiving the failure notification, relevant personnel should track and resolve the issue in a timely manner. Establish an issue tracking system to record the progress and results of the issue to ensure that the issue can be resolved in a timely and effective manner.

3.6 Summary and Optimization

After completing the testing, the testing process is summarized to analyze the common failure reasons and suggest optimization. This not only improves the efficiency of future testing, but also ensures the continued effectiveness of the go-live strategy.

Through such a process, it can ensure the comprehensiveness and effectiveness of the testing methodology and provide a solid foundation for subsequent strategy optimization and system improvement.

4 Analysis of Experimental Results

In the course of testing and running the system, we identified the following major issues:

4.1 Concurrent Disposition Value Override Problem

When the policy system disposes of the status of the article being commented on while labeling the comments under the article accordingly, due to concurrent dispositions, there may be a situation in which the dispositions are overwritten. This indi-

cates that in the system design, the control of concurrent operations is not strict enough, and no effective locking mechanism or transaction management is adopted to ensure data consistency. To solve this problem, the following measures can be considered:

pessimistic lock: Before updating data, lock the relevant data to ensure that other processes cannot modify them during the locking period. You can use a database row lock or table lock, or implement distributed locking at the application layer[7].

Optimize transaction processing: For disposal processes involving multiple operations, use database transactions or distributed transactions to ensure the atomicity and consistency of operations.

4.2 The Disposal of Unsuccessful Problems

There are occasional inconsistencies when updating the cache, resulting in the state being changed to 0, thus making the disposal unsuccessful. This may be due to defects in the cache update mechanism, or network problems, system failures and other anomalies in the update process. There are ways to solve this problem:

Enhance cache consistency management: You can use cache update strategies, such as updating the database before updating the cache, or use the cache invalidation mechanism to ensure that the data in the cache is consistent with the data in the database.

Increase the reliability of cache updates: You can use a distributed cache system to improve the availability and reliability of the cache. At the same time, you can set up a retry mechanism for cache updates to automatically retry when there is an update failure to ensure the correctness of the cached data.

4.3 The Delay in Playing Feature Interface for Users

The delay in hitting the user feature interface leads to the data hit when the user features can not be found, do not meet the policy requirements do not hit. This problem may be caused by poor interface performance, network delays or high system load. To solve this problem, the following measures can be taken:

Optimize interface performance: optimize the performance of the interface for hitting user characteristics, such as reducing the number of database queries, optimizing algorithms, etc., to improve the response speed of the interface.

Increase caching: for frequent queries of user characteristics data, you can set up a cache to reduce the number of calls to the interface and improve query efficiency.

Monitor and optimize system performance: Monitor the load of the system in real time and optimize the performance bottleneck to ensure that the system can run stably and efficiently.

4.4 The Comment Status is Not Disposed of Problem

The platform's disposal cache is inconsistent, the update of a server room fails, and the showbatch interface request goes to the failed server room and the server room

has not retried, resulting in the comment status not being disposed of. This problem is mainly due to cache synchronization problems between server rooms. The solution is as follows:

Improve the server room cache synchronization mechanism: Ensure that the cache between each server room can be synchronized in a timely manner to avoid inconsistencies. You can use distributed cache synchronization techniques, such as master-slave synchronization of Redis clusters or distributed cache middleware.

Increase the machine room retry mechanism: when an update fails in a machine room, it should be automatically retried to ensure that the data can be correctly disposed of. At the same time, the number of retries and the time interval can be set to avoid unlimited retries leading to system performance degradation.

Monitor the status of the server room: real-time monitoring of the status of each server room, timely detection and treatment of server room failures to ensure system stability and reliability.

The proportion of the number of occurrences of the above four problems is shown in Fig.1 below.

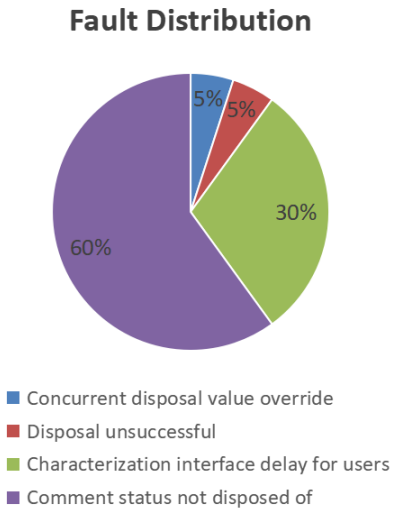

Fig. 1. Fault distribution map

5 Conclusions

The testing methodology presented in this paper is of great value in practical applications. By writing code to read the policy rules in the online repository, especially the parsing of drools rules, a solid foundation is laid for the assembly and validation of subsequent use cases.

In the process of design and implementation, several key factors such as comment publisher information, users in the account pool, comment contents, exemption rules and disposition methods are fully considered, so that the assembled use cases are

highly targeted and effective. The use case validation session ensures the accuracy and stability of the system by comparing the actual state with the expected results. The email notification mechanism for failed use cases identifies problems in a timely manner and prompts the relevant personnel to take prompt action to resolve the problem.

However, the testing methodology is not perfect. In practice, it faces the challenge of code maintenance, which requires constant efforts to ensure the validity of the code as the online library changes and rules are updated. At the same time, the complexity of the testing methodology demands a high level of skill from developers and testers, which also puts a higher demand on team training and collaboration. In addition, the accuracy of the data in the account pool is critical to the quality of the use cases, requiring an effective data quality monitoring mechanism.

Looking ahead, we will continue to optimize and improve this testing methodology. Continuously enhance the maintenance and updating of the code to adapt to the changing business requirements. Improve the ability to cope with complex testing scenarios through enhanced team training and collaboration. Further improve data quality monitoring to ensure the accuracy and reliability of use cases. We believe that with continuous efforts, the testing methodology will provide more stable and efficient guarantee for the conference-related systems and businesses.

References

1. TUFCHI S., YADAV A, AHMED T. (2023)A Comprehensive Survey of Multimodal Fake News Detection Techniques: Advances, Challenges, and Opportunities. International Journal of Multimedia Information Retrieval, 12:28-36. doi:10.1007/s13735-023-00296-3.
2. QIU X., SUN T., XU Y., et al. (2020)Pre-trained Models for Natural Language Processing:A Survey. Science China(technological sciences), 63:1872-1897. https://d.wanfangdata.com.cn/periodical/zgkx-ee202010001.
3. Grefenstette E., Blunsom P. (2014)A convolutional neural network for modelling sentences. In: The 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland. pp. 655-665. https://doi.org/10.48550/arXiv.1404.2188.
4. CHO K., MERRIENBOER B., GULCEHRE C., et al. (2014)Learning phrase representations using RNN encoder-decoder for statistical machine translation. https://arxiv.org/pdf/1406.1078
5. Kang S. A., Kim Y. S., Choi S. H. (2015)Study on the Social Issue Sentiment Classification Using Text Mining. Journal of the Korean Data and Information Science Society, 26: 1167-1173.
6. LIANG Y., TOHTI T., HAMDULLA A. (2023)Multi-modal False Information Detection via Multi-layer CNN-based Feature Fusion and Multi-classifier Hybrid Prediction. Computer Engineering and Science. 45:1087-1096. http://joces.nudt.edu.cn/EN/abstract/abstract17680.shtml.
7. Narger D., Lehman E., Leighton T., et al. (1997)Consistent hashing and random trees:Distributed caching protocols for relieving hot spots on the World Wide Web. In: Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, EL Paso, 654-663. https://dl.acm.org/doi/10.1145/258533.258660.