A Quality Detection System for Book Reviews Posted on Goodreads

Lauren Rutledge

July 2025

Project Files and Code available at:
`https://github.com/laurenrutledge/goodreads-detection-system`

Abstract

This project presents the foundation of an automated system for assessing the quality of free–text book reviews collected from Goodreads. Leveraging a large, publicly available dataset of user reviews, interpretable linguistic and metadata features were extracted to train a multinomial logistic regression model that classifies reviews into ”quality levels”.

This metadata–driven baseline was selected initially due to practical time and compute constraints, offering a fast, interpretable, and resource–efficient method to begin exploring review quality at scale. Ongoing work is now focused on integrating transformer–based models, starting with a BERT (Bidirectional Encoder Representations from Transformers) architecture, to directly analyze raw text for semantic depth, coherence, and contextual richness. With more time, future directions (which are also currently being explored) include incorporating human–labeled data to further refine model performance and experimenting with additional NLP architectures. A primary objective of this project, nevertheless, is to transition toward advanced NLP methods, as such methods represent a critical step toward building a more robust, scalable, and nuanced review quality detection system.

1 Background

Goodreads – a widely used online book review platform – was founded in 2007 by Otis and Elizabeth Khuri Chandler who were on a simple mission: to help people discover, share, and write reviews on books they love. By 2019, the platform had accumulated over 90 million user-generated reviews. [1] Today, Goodreads hosts a database of more than 3.5 billion books and millions of reviews that not only drive book discovery but also shape public opinion and reading trends worldwide. [3]

While online free-form book review platforms like Goodreads have become highly influential, it is increasingly clear that the sheer volume of content does not guarantee meaningful insight. Buried within millions of entries are countless reviews that do little more than clutter the platform—let alone those laced with malicious links, spam, and / or self-promotion masquerading as feedback. Moreover, many of these user-written reviews come with inconsistencies, informal language, or ratings that conflict with the accompanying text, resulting in posts that offer little or no real substance. Far from fostering a community of engaged readers, these low-quality contributions can drown out thoughtful voices, distort ratings, and erode the very trust that makes such platforms valuable in the first place!

In this project, I set out to confront this challenge by analyzing Goodreads' free-text review data with the goal of detecting and categorizing review quality issues. Using a combination of rule-based heuristics, feature engineering, and natural language processing (NLP), I explored patterns in free-text to uncover the most critical anomaly types such as rating–text mismatches and overly short or low-information reviews.

The result is a system that (1) automatically flags malicious or overtly problematic reviews and (2) distinguishes and elevates more informative, substantive reviews over low-quality content. By filtering out noise and prioritizing meaningful contributions, this system strengthens the reliability of Goodreads data and lays a foundation for cleaner, more trustworthy free-text review ecosystems.

1.1 Problem Statement

The problem addressed in this project is to design and implement a system that can automatically detect and categorize review quality issues in Goodreads' free-text data. Specifically, the system developed in this project must:

1. Identify and flag malicious or inappropriate book reviews, and
2. Highlight and prioritize high-quality, informative contributions.

In doing so, the system aims to improve the integrity, usefulness, and overall reliability of Goodreads' free-text review data.

Furthermore, while this Data Quality Detection System is specifically tailored to Goodreads in its initial iteration, it is designed to be scalable and adaptable for use across other popular book-review platforms (and beyond!) in the future.

2 Project Scope, Dataset and Assumptions

2.1 Scope

Due to the time and computational constraints of this project, the current implementation focuses exclusively on analyzing and evaluating the quality of user-generated book reviews within the Goodreads platform. However, a strong motivation for future work is to scale this quality detection system to other book-review platforms, and eventually to broader product-review platforms that support free-text reviews. Additionally, the scope is intentionally limited to free-text reviews and their associated metadata only, as this project is solely focused on developing a system that evaluates the quality of free-text data.

The system is designed to detect and categorize quality issues, which are organized into two tiers:

1. **Tier One:** Reviews containing malicious content, such as harmful links, spam, or self-promotional material.
2. **Tier Two:** Reviews that exhibit an overall quality score. This score is determined by combining calculated scores within the following two categories:

(a) **Substantiveness Score:** Measures the depth and informativeness of the review.

(b) **Appropriateness Score:** Evaluates whether the review content aligns with platform guidelines and the associated rating.

In short, Tier Two also encompasses structural or informational anomalies, such as rating–text mismatches and extremely short or low-information reviews.

Finally, the current iteration of this project does not incorporate multimedia content (e.g., images), non-English reviews, or real-time moderation pipelines. Instead, it focuses on a proof-of-concept system capable of operating on an existing dataset of Goodreads reviews. As such, future work also includes expanding this system to monitor reviews written in languages beyond English, as well as exploring additional data modalities and real-time monitoring abilities.

2.2 Data

For practical and reproducible development, I chose to work with existing Goodreads review datasets made available through prior research rather than collecting new data from Goodreads in real time. Specifically, I relied on datasets of detailed user reviews introduced in the following papers: [4], [5]. While future work could extend to collecting public reviews from Goodreads in real time (Goodreads.com receives approximately 45 million active visitors per month), for this project I utilized datasets curated and published by Julian McAuley, which are openly available through his GitHub repository and dataset portal [2].

For background, McAuley's published datasets were collected in late 2017 from goodreads.com and are a result of scraping and consolidating users' public shelves, i.e. shelves where everyone can see it on the web without login. This dataset also keeps user IDs and review IDs are anonymized. Of the datasets published, for this project, only the ”Users' Detailed Book Reviews” json was used. Today, these data sets, including ”Users' Detailed Book Reviews”, can be downloaded by navigating to the gz file entitled: `goodreads_reviews_dedup.json.gz` within McAuley's repository [2].

After downloading the raw json from McAuley, the data was cleaned by removing all columns except the following:

- **user_id**: An anonymized identifier for the user who wrote the review.
- **review_id**: An anonymized identifier for the specific review entry.
- **review_text**: The free-text content of the user's review.
- **rating**: The user's star rating for the book (typically an integer from 1 to 5).
- **date_added**: The timestamp indicating when the review was posted.
- **n_votes**: The number of votes or "helpful" marks the review has received from other users.

Next, because this project's goal was to focus on the quality of free-text data, any entries with an empty **review_text** field or those containing only blank spaces / line breaks were removed from the dataset. Additionally, entries missing data for **user_id**, **review_id**, or **date_added** were also removed during the data cleaning phase.

Finally, as mentioned above, the scope of this project was limited to reviews containing free-text in English, which was determined using the Python library `langdetect`.

2.3 Assumptions

The following assumptions were made in order to scope and implement this project within the given time and computational constraints:

- **Data Availability:** The datasets curated and published by Julian McAuley are assumed to be accurate, complete, and representative of Goodreads' public review ecosystem.
- **Metadata Reliability:** Fields such as **user_id**, **review_id**, **rating**, and **date_added** are assumed to be correctly associated with each review and free from systemic errors.
- **Static Dataset:** The analysis is based on a static dataset collected in 2017, and it is assumed that trends and anomalies detected remain relevant to the current Goodreads platform.
- **Scope of Content:** Only free-text reviews are considered. Any data entry that does not contain a free-text review was excluded from the scope. Multimedia content (images, videos) and non-review interactions (likes, comments, etc.) are outside the scope of this project.
- **Data Language:** All reviews analyzed are in English, or in ”English slang”. Reviews written in other languages were removed, based on language detection using the `langdetect` library.
- **Link Presence Indicates Spam or Malicious Content:** Any review containing a hyperlink (e.g., starting with `http://`, `https://`, or including patterns such as `www.`, `.com`, `.org`, or `.net`) is assumed to be spam, promotional content, or potentially malicious. These reviews are therefore flagged and considered low-quality, regardless of other factors.

2.4 Target Users and Key Players

To further define the scope of this project, it is important to identify the intended users and stakeholders of this Data Quality Detection System.

Within Goodreads, the primary end-users and key players who would find this system highly valuable include:

- **Content Moderators:** Responsible for reviewing and removing malicious, inappropriate, or low-quality reviews.

  Content moderators are typically full-time employees who must sift through hundreds of reviews each day, often under time pressure. Because individual moderators may judge quality differently, a system like this one can provide a standardized pre-analysis summary and rank reviews by severity, allowing moderators to focus their attention where it is most needed.
- **Platform Managers:** To maintain the overall integrity of the review ecosystem and inform decisions about platform policies and features.

  Included in ”Platform managers” are classic product managers overseeing review systems as well as business stakeholders concerned with user engagement and long-term platform health. Platform managers often lack timely visibility into text review trends and may only be alerted when functional issues escalate. A system like this can surface prioritized quality insights, highlight emerging book-specific trends, and present actionable user-feedback data to guide business strategic decisions.

- **End-Users (Review Readers):** To benefit from a cleaner, more trustworthy set of reviews when making book selections.

  Book enthusiasts looking for reading recommendations (or just a book enthusiast wanna-be, such as myself) face an overwhelming volume of reviews, many of which vary widely in helpfulness. These users are in need of a system that prioritizes high-quality, substantive reviews to help readers navigate this information overload, make more confident book selections, and fall in love with the book-loving community.

- **Authors:** To gain more reliable feedback and insights from genuine, high-quality reviews.

  By filtering out spam and low-effort comments, this system allows authors to focus on meaningful reader feedback that can guide their future work and strengthen their engagement with audiences.

3 Methods: Feature Engineering and Models

3.1 Feature Engineering

Since the objective of this project is to design a system that automatically detects and categorizes review quality issues in Goodreads' free-text data, a preliminary exploratory data analysis (EDA) was conducted to identify measurable indicators of review quality that could be extracted directly from the review text and its associated metadata. This goal of this step was to derive a set of features that are both computationally efficient to calculate and predictive of review substanstiveness, appropriateness, and overall quality.

For Tier One, it was determined whether or not each review contained a link via typical link regex formatting. If the review had a link within the free-text, the review was flagged. If not, the review was able to move on to Tier Two for quality labeling.

For Tier Two, the following engineered features were selected based on their statistical distribution in the dataset and their potential to reflect depth and informativeness:

- **Sentence Count**: the total number of sentences detected in the review, computed using NLTK's sentence tokenizer. Higher values tend to indicate richer content and more elaboration.

- **Word Count**: the total number of word tokens in the review, as a proxy for review length and potential detail.

- **Average Words per Sentence**: the ratio of word count to sentence count, capturing sentence complexity and potential informational density.

- **Lexical Diversity**: the type–token ratio (unique word count divided by total word count), reflecting vocabulary variety and potential thoughtfulness in expression.

- **Number of Votes (*n_votes*)**: the total number of votes a review received from other Goodreads users, used as a social signal of perceived quality or relevance.

In addition to these core features, several interaction terms were generated to capture combined effects in Tier Two as well, such as:

- **Sentence–Word Interaction**: a product term between sentence count and word count, designed to capture the joint impact of length and structure.
- **Sentence–Average Words Interaction**: combining sentence count with sentence complexity to highlight reviews that are both lengthy and dense.
- **Lexical–Sentence Interaction**: an interaction between lexical diversity and sentence count, identifying reviews that are both long and lexically varied.
- **Words per Sentence Ratio and Unique Words per Sentence**: additional derived ratios to provide normalized measures of richness per unit of structure.

It should also be noted that all continuous features were standardized to zero mean and unit variance prior to model training. This was done to ensure that differences in scaling would have no effect on a single feature dominating the learning process. Ultimately, it is / was important that the coefficients of the logistic regression remain interpretable in terms of relative influence.

3.2 Model Usage

To predict the "quality level" of each Goodreads review, a multinomial logistic regression classifier was employed first. In short, logistic regression is a linear model that estimates the conditional probability distribution over a discrete label space given an input feature vector.

In this case, the number of labels used to rank quality in the free-text reviews was five. All reviews assigned the label: 1 represent the lowest-quality reviews, and the label 5 indicated the highest quality reviews. This lead to a multi-class set-up with $K = 5$ classes, meaning the model learns a weight matrix $W \in \mathbb{R}^{K \times d}$ and bias vector $b \in \mathbb{R}^K$ such that for an input feature vector $x \in \mathbb{R}^d$, the probability of class k is given by the softmax transformation:

$$P(y = k \mid x) = \frac{\exp(w_k^\top x + b_k)}{\sum_{j=1}^{K} \exp(w_j^\top x + b_j)},$$

where $w_k^\top$ is the k^{th} row of the weight matrix.

Logistic regression, paired with a training-test split of 80-20, was selected as the baseline model because of its interpretability, robustness on tabular data, and above all, due to it's computational efficiency, which was necessary for the scope of this project. Given that the feature space is composed of well-engineered numeric indicators, a linear decision boundary was considered sufficient to capture meaningful relationships between features and the substantiveness and appropriateness labels.

Unlike more complex models, logistic regression allows direct inspection of learned coefficients, enabling a clear understanding of how each feature contributes to the classification decision. And most importantly in this scenario, it trains quickly on large datasets and provides a strong, well-understood baseline against which more sophisticated models (e.g., fine-tuned BERT) can be compared. Indeed, a BERT model is currently being set up for such comparisons and to expand on the project!

3.3 Measurable Outcomes

The performance of the proposed logistic classifier was quickly evaluated and engineered towards a "Goodreads Evaulation System" using standard multi–class classification metrics. Specifically, the following measurable outcomes were derived from the logistical regression model:

- **Accuracy**: the proportion of correctly predicted labels over all test examples.
- **Precision**: for each class k, the ratio of true positives to all instances predicted as class k. Precision captures the model's ability to avoid false positives.
- **Recall**: for each class k, the ratio of true positives to all actual instances of class k. Recall captures the model's ability to identify all relevant examples.
- **F_1 Score**: the harmonic mean of precision and recall for each class, providing a single metric that balances both false positives and false negatives.
- **Confusion Matrix**: a $K \times K$ matrix that summarizes the counts of true vs. predicted labels, enabling detailed inspection of class–specific performance.

These metrics are computed on a held–out test set after model training. The use of multiple metrics ensures a comprehensive assessment of model behavior, beyond overall accuracy, by capturing class–level trade–offs between precision and recall.

4 Results and Discussion

4.1 Results

The complete experimental results, including accuracy, precision, recall, F_1 scores, and confusion matrices, are documented in detail within the project repository. All code, intermediate data artifacts, and Jupyter notebooks used to train, evaluate, and analyze the logistic regression model are publicly available on GitHub:

`https://github.com/laurenrutledge/goodreads-detection-system`

This repository contains reproducible notebooks for feature engineering, model training, and evaluation, along with instructions for running the system locally.

4.2 Discussion

The trained logistic regression model serves as the core of a prototype system designed to automatically assess the quality of Goodreads free–text reviews. In a production setting, this model could be deployed as part of a multi–tiered review moderation pipeline, where new incoming reviews are passed through the following stages:

1. **Preprocessing and Feature Extraction:** Each raw review is first cleaned and transformed into the engineered feature space (e.g., sentence counts, lexical diversity).

2. **Automated Classification:** The standardized features are fed into the trained classifier, which outputs a predicted substantiveness level.

3. **Action Routing:**

   - Reviews predicted as high priority are surfaced to the top of a recommendation feed or highlighted to moderators.
   - Medium and low priority reviews are retained but ranked lower in visibility.
   - Reviews flagged as low quality or suspicious (e.g., spam) are queued for removal or human verification. Overtime, those flagged for suspicion can be removed in the tenths of a second between the end-user clicking on "submit the text-review" and the review being posted to the public forum.

5 Current and Future Work

While the current system establishes a strong metadata–driven baseline, further methods are currently being carried out to improve and advance this project. Precisely, the following items are underway now (and in this order!):

- **Integration of Transformer–Based Models:** Work is currently underway to develop and fine–tune a BERT model (Bidirectional Encoder Representations from Transformers) first. The resulsts of this model can complement or even replace the current feature–engineered logistic regression approach. Unlike metadata–based features, a BERT model can directly capture nuanced semantic patterns, contextual coherence, and latent relationships within the raw text, providing deeper insight into review quality. This step is a major priority, as transformer architectures are non-negotiable in this day and age. Following the BERT model, other NLP models will be carried out as well. and will likely yield substantial improvements in accuracy and robustness.

- **Coherence Scoring:** Beyond simple counts and diversity measures, future iterations of the logistic regression classifier will incorporate a learned or algorithmic coherence score. Such a metric can evaluate the logical flow and sentence connectivity of a review, ensuring that even high word counts reflect meaningful substance rather than fragmented or incoherent content.

- **Plagiarism and Malicious Content Detection:** The system will be extended with a plagiarism detection module that flags reviews with identical or near–identical text authored by different users. Such duplicated content can degrade trust in the platform and skew recommendation signals. By leveraging techniques such as cosine similarity on embedding vectors or hashing–based duplicate detection, the system will distinguish between genuinely original reviews and likely copy–pasted or malicious entries.

- **Iterative Model Improvements:** Future development will include active learning pipelines where human feedback on flagged reviews is incorporated to iteratively refine both the metadata–driven and transformer–based models.

References

[1] N Ghugare. Book recommendation system using goodreads dataset. 2024.

[2] Julian McAuley. Goodreads datasets. `https://cseweb.ucsd.edu/~jmcauley/datasets/goodreads.html\#datasets`, 2025. Accessed: July 2025.

[3] Shadi Shahsavari, Ehsan Ebrahimzadeh, Behnam Shahbazi, Misagh Falahi, Pavan Holur, Roja Bandari, Timothy R. Tangherlini, and Vwani Roychowdhury. An automated pipeline for character and relationship extraction from readers literary book reviews on goodreads. com. In *Proceedings of the 12th ACM Conference on Web Science*, pages 277–286, 2020.

[4] Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*, pages 86–94. ACM, 2018.

[5] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*, pages 1909–1919. Association for Computational Linguistics, 2019.