Real-Time Event Recognition in MLB Social Media Streams

Brandon L Medina

Claremont Graduate University

IST 332: Natural Language Processing

Hengwei Zhang, PhD

December 10, 2025

1 Introduction

1.1 Problem Statement

In the game of baseball there is no shortage of structured data, but fan reactions on social media remain unstructured and underutilized in content generation or even broadcast. This limits a team's ability to track engagement, identify key moments, and assess sentiment in real-time. Using NLP models to align social comments with the large amount of available data can convert unstructured discourse into insights, bridging quantitative game data and audience reactions.

1.2 Research question

This research aims to answer the question: How effectively can machine learning and natural language processing (NLP) models detect and classify Major League Baseball (MLB) in-game events in real time using temporally aligned social media comments?

1.3 Sub Questions

To achieve this research question, I broke down the significant overarching research question into three main milestone questions:

- To what extent can temporally aligning Reddit Game Thread comments with official MLB play-by-play data be used to label fan discourse for in-game event classification?
- How do classical NLP models using TF-IDF features compare to transformer-based models (DistilBERT) in classifying MLB in-game events from short, informal fan comments?
- Can models trained on comments with known outcomes effectively generalize to unseen games or new social media platforms (e.g., X/Twitter)

1.4 Hypothesis

Thinking about the questions above, I hypothesize that NLP can accurately identify and classify MLB in-game events from fan-generated comments when aligned with official play-by-play data.

1.5 Targeted Variables

- Relevance classification – determines whether a comment refers to an actual in-game event.

- Sentiment Classification – determine general fan sentiment towards plays or calls in game.
- Event classification – identifies the specific event that occurred in-game (e.g., homerun, strikeout, stolen base) based on the comment text.

2 Corpus Creation

3 Introduction

Our goal was to create a medium-sized, temporally aligned text dataset linking fan commentary and MLB play-by-play data. The corpus needed 10,000 records, extensive metadata, and approximate labels with in-game events. Reddit Game Thread discussions from r/Baseball were paired with MLB Statcast data to form the dataset.

3.1 Reddit Comment Collection

3.1.1 Data Source & Collection

I chose Reddit's r/baseball "Game Thread" posts because they have high-volume, real-time fan reactions during games and event-focused comments referencing pitches, hits, and outs. These threads are ideal since they focus discussion to a specific game. Comments were collected via PRAW by identifying official "Game Thread" posts through title matching, filtering for dates, and prioritizing top comments to capture reactions.

3.1.2 Results

The Reddit collection process gathered about 17,500 comments from seven World Series Game Thread posts. Each comment was stored in a CSV file, and to ensure ethical use, all usernames were removed and only anonymized identifiers were used.

3.2 MLB Play-by-Play Collection

3.2.1 Data Source & Collection

Official MLB play-by-play data was obtained using the pybaseball library, which provides access to Statcast pitch-level data. This source was chosen because it offers detailed event descriptions (such as balls, strikes, hits, outs), maintains consistent identifiers across games, and has a structured format compatible with timestamped comments. Statcast data is the definitive reference for in-game events, helping convert fan reactions into weakly labeled training data. The collection pipeline involved querying Statcast data for the postseason date range, identifying World Series game_pk values, retrieving pitch-level records for each game via statcast_single_game, and combining all games into a unified dataset with pitch order, event descriptions, and game context.

3.2.2 Results

The dataset comprised roughly 2,400 pitch-level records, including key fields such as: game_pk (a unique game identifier), inning and pitch_number (indicating the pitch sequence), description (detailing the pitch outcome), and events that depict the higher-level outcome (e.g., home_run, strikeout).

3.3 Temporal Alignment of Comments and Events

1. Alignment Strategy - To connect fan commentary with gameplay, a weak supervision approach based on temporal proximity was used. The process involved several stages:
2. Timestamp Standardization - Reddit comment timestamps were converted to UTC. Official game start and duration times were verified and converted from local stadium time zones.
3. Game Window Filtering - Comments were limited to a specific in-game window (from 3 minutes before the first pitch to 5 minutes after game end) to exclude pregame hype and postgame chatter.
4. Estimated Pitch Timestamp Reconstruction - Since Statcast timestamps were unavailable, pitch times were calculated using pitch order, inning transitions, MLB pitch-clock assumptions (15-20 seconds), and delays from inning breaks and pitching changes.
5. Nearest-Neighbor Matching - Each comment was matched to the nearest estimated pitch event using a merge_asof with a ±300-second tolerance. Unmatched comments were excluded from modeling.

6. Result of Alignment - This produced a dataset in which many in-game comments received approximate event labels. Although not highly precise, the alignment was adequate for supervised learning with weak labels.

3.4 Final Corpus Output

After filtering, the final modeling corpus consisted of 15,173 matched records used in this project. The data was anonymized by dropping the username column for ethical reasons. The dataset was saved as CSV files and used for preprocessing, exploratory analysis, sentiment analysis, topic modeling, and supervised learning. This process meets the project's scale, structure, and reproducibility requirements and also reflects the challenges of working with real-time, user-generated social media data.

4 Text Cleaning and Preprocessing

4.1 Preprocessing foundation

Before applying linguistic normalization, the dataset was filtered to include only comments that successfully matched a recognizable MLB event. Entries lacking event descriptions were removed, ensuring each remaining comment had a weak label suitable for supervised learning. This step reduced label noise and prevented models from training on comments unrelated to gameplay. Additionally, comments that were mostly non-textual, such as GIF placeholders or embedded media references, were excluded. After filtering out unmatched events and GIF-based comments, 966 event-linked comments were discarded, leaving a clean, reliable dataset ready for further preprocessing and modeling.

4.2 Lowercasing

All comment text was converted to lowercase to standardize lexical variations and reduce vocabulary sparsity. This process ensures that semantically identical words (e.g., "Strike," "strike," "STRIKE") are treated as a single token, improving consistency across tokenization, stopword removal, and feature extraction. Lowercasing simplifies vocabulary and stabilizes both traditional NLP models and transformer-based tokenization.

4.3 Removal of URLs, Markdown Artifacts, Emojis, and Non-Linguistic Noise

Non-linguistic elements such as URLs, Reddit Markdown syntax, GIF/image links, and emojis were eliminated from the text. These artifacts do not contribute to event classification and might

inflate vocabulary size or introduce noise tokens. Emojis were converted to Unicode, and excessive whitespace was normalized. This process ensured that only relevant textual content proceeded to subsequent preprocessing steps.

4.4 Removal of Punctuation and Special Characters

Following noise removal, regular expressions were used to delete all punctuation and non-alphabetic characters. This process guaranteed that tokens contained only alphabetic characters, preventing punctuation marks or other symbols from forming separate tokens. Normalizing spacing and removing special characters prepared the text for accurate and uniform tokenization.

4.5 Tokenization

Tokenization was performed using a simple whitespace-based method, splitting each cleaned comment into individual word tokens. Given the short and informal style of Reddit comments, this method was sufficient for capturing lexical units without unnecessary complexity. This approach at the tokenization stage enabled effective stopword removal and lemmatization while maintaining computational efficiency.

4.6 Stopword Removal

Common English stopwords were removed using spaCy's predefined list. These frequent function words (like "the," "and," "is") add little value for event classification and often overshadow more meaningful features. Eliminating stopwords reduced token counts and emphasized content-rich terms that better distinguish MLB event types.

4.7 Lemmatization

Lemmatization was performed using spaCy's lightweight English model to convert tokens to their root forms. This process unified the inflected variants of a single word (such as "hits," "hitting," and "hit") into a single base form, improving lexical consistency and reducing sparsity. Lemmatization was performed after removing stopwords to boost efficiency and concentrate on meaningful tokens. The resulting lemmas were compiled into a final clean_text field for modeling.

4.8 Corpus Statistics

Following preprocessing, summary statistics were generated to describe the cleaned corpus. The average length of raw comments was 69.64 characters or 13.11 tokens, highlighting the

conversational style of Reddit Game Threads. Post-cleaning, the average comment length dropped to 6.84 tokens, reflecting the successful elimination of noise and unnecessary information. The final vocabulary comprised 1,612 unique lemmas, with a lexical diversity (type–token ratio) of 0.2441, indicating a concise yet rich vocabulary. These metrics demonstrate that the preprocessing pipeline effectively minimized noise while preserving key linguistic signals for subsequent NLP analysis.

5 Data Understanding and Preparation

5.1 Distribution of Comment Lengths

Comments are brief, averaging 6.84 tokens with a median of 6, mainly concise reactions. The interquartile range (4–9 tokens) indicates limited content, with some outliers reaching 57 tokens. This underscores the need for models suited for short, informal texts, validating the use of TF-IDF and transformer models.

5.2 Word Frequency Analysis

Analysis of lemmatized text shows common terms like game, hit, run, pitch, and ball, alongside player and team names such as Ohtani, Dodgers, and World Series. Casual markers indicate spontaneous fan comments. The data primarily reflects MLB gameplay, is useful for event classification, and includes typical social media noise.

5.3 Event Frequency Distribution

MLB event labels are imbalanced: common outcomes include ball, hit_into_play, swinging_strike, foul, and called_strike; rare events like hit_by_pitch and foul_bunt occur less frequently. This mirrors baseball rates, but challenges supervised learning due to class bias.

5.4 Identified Data Issues and Mitigation Strategies

Reddit comments rarely mention the pitch directly, and estimated timestamps add uncertainty. Comments were limited to three to five minutes before or after the first pitch, excluding unmatched data. To address class imbalance, labels were grouped into broader categories, and class-weighting during training improved model stability.

6 Sentiment Analysis

6.1 Motivation and Hypothesis

Sentiment analysis was conducted to determine whether fan emotional responses vary significantly across different in-game events and whether sentiment can serve as a useful additional indicator. The primary hypothesis was That Fan sentiment generally becomes more positive during positive offensive events, such as hits and home runs, and more negative during negative events, such as strikeouts and routine outs. Testing this hypothesis helps understand whether sentiment trends align with gameplay outcomes and if sentiment features can enhance event classification models.

6.2 Sentiment Analysis Method

Sentiment scores were calculated using VADER, a rule-based analyzer optimized for short, informal social media text. VADER was chosen because Reddit comments are brief, conversational, and expressive, fitting VADER's lexicon. Sentiment analysis provided four metrics: positive, neutral, negative, and compound scores. The compound score, which indicates overall sentiment, was used for comparisons across events and time.

6.3 Sentiment Analysis

Average sentiment scores for each MLB event showed minor differences, with offensive plays slightly higher and strikeouts lower, all close to neutral. This suggests fan comments are mainly conversational rather than emotionally charged, reflecting engagement. Sentiment Trends Over Time: Rolling-average analysis revealed consistent sentiment throughout games, with brief fluctuations linked to activity clusters rather than long-term emotional shifts. No persistent sentiment spikes aligned with in-game events indicate a steady emotional tone in fan discussions.

6.4 Hypothesis Testing and Results

To formally test the hypothesis, events were grouped into broader sentiment categories: positive offensive outcomes and negative or neutral outcomes, such as strikeouts and routine outs. The mean compound sentiment scores of these groups were compared using an independent t-test. Although positive offensive events showed a marginally higher average sentiment, this difference was not statistically significant ($p > 0.05$). Therefore, the data did not support the hypothesis that fan sentiment differs significantly by event type.

6.5 Interpretation and Implications

The sentiment analysis results suggest that sentiment alone is not sufficient to identify specific MLB in-game events in real-time Reddit discussions. Fans tend to react more to the overall game context and ongoing conversations than to individual pitch results. Consequently, sentiment was not a primary feature in supervised event classification models. Nonetheless, sentiment analysis offers valuable insights into overall fan engagement and emotional tone, even if it does not directly improve event prediction accuracy. These findings support the idea that linguistic content and contextual meaning are more effective for classifying events than emotional polarity alone.

7 Topic Modeling

7.1 Purpose and Motivation

Topic modeling explored recurring themes in Reddit Game Thread discussions to gain deeper insight into fan discourse beyond sentiment and word frequency. While sentiment analysis focused on emotional polarity, topic modeling revealed discussion patterns and tested alignment with gameplay or events.

7.2 Modeling Approach

LDA was chosen for its interpretability and effectiveness in short-text aggregation. It was applied to cleaned, lemmatized comments with a bag-of-words approach. Various configurations were tested, including topic counts (5, 8, 10), n-gram ranges, and minimum document frequency thresholds. Model performance was evaluated using topic coherence (c_v), which measures semantic consistency among top words. The best configuration—five topics with optimized vectorization—was selected for further analysis.

7.3 Discovered Topics and Interpretation

The final LDA model revealed five primary conversational themes: Emotional Reactions and Disputes, covering feelings like excitement, frustration, or disagreement; Team Support and Momentum, which involves comments on team performance and game flow; Player Performance and Critique, focusing on individual players and strategies; At-Bat and Pitch-Level Reactions, capturing immediate responses to pitches and swings; and Game-Level Context and Storylines,

discussing the broader narrative of the World Series. These themes showcase the variety of fan conversations, blending emotion, tactics, and storytelling.

7.4 Topic Distribution and Patterns

Analysis showed comments were spread across themes, mainly at-bat reactions and emotions. Topics didn't align with MLB labels; individual events appeared in multiple topics, indicating discussion focuses on framing rather than specific outcomes. This suggests Reddit Game Threads are narrative-driven, with fans emotionally reacting, evaluating, and placing events in context.

7.5 Limitations and Implications

Topic modeling identified meaningful themes, but it isn't ideal for direct event classification. It shows how fans discuss the game, not the specific events, so labels weren't used as predictors. However, it offers useful insights into fan engagement and supports that event classification depends more on contextual semantics than on thematic structure. These findings align with the presentation results, indicating that unsupervised methods are better suited to exploratory analysis than to precise event detection.

8 Supervised Learning and Model Evaluation

8.1 Objective and Learning Setup

This project evaluated how well NLP models classify MLB in-game events from Reddit comments. Raw Statcast labels are granular and imbalanced, so outcomes were grouped into broader categories like hit, walk, strikeout, and out. This improved class balance and suited the noisy, brief fan commentary. The task tests if fan language can predict gameplay outcomes under weak supervision.

8.2 Feature Engineering

Text features were created using TF-IDF vectorization with unigrams and bigrams, a document frequency threshold, and stopword removal, making them ideal for sparse, high-dimensional text like Reddit comments. For transformer models, cleaned text was tokenized with DistilBERT’s WordPiece for contextual embeddings.

8.3 Train–Test Split and Evaluation Metrics

The dataset was split using methods to preserve class distribution. Performance was then evaluated using accuracy and macro-averaged F1 score, with focus on Macro-F1 due to class

imbalance. This metric ensures minority categories contribute equally and prevents scores from being inflated by majority classes.

8.4 Classical Model Performance

8.4.1 Logistic Regression

Logistic Regression served as a linear baseline, achieving an accuracy of 0.544 and a Macro-F1 score of 0.292. The model demonstrated reasonable balance across event categories but was limited by its reliance on surface-level TF-IDF features. While it captured obvious lexical cues associated with common events, it struggled to distinguish subtler semantic differences in short and informal fan comments, resulting in moderate overall performance.

8.4.2 Linear Support Vector Machine (SVM)

The Linear SVM outperformed other classical models in terms of class-balanced performance, achieving an accuracy of 0.606 and the highest Macro-F1 score among TF-IDF-based models (0.303). Its margin-based optimization allowed it to better separate overlapping event classes in sparse, high-dimensional feature space, making it more robust to noisy labels introduced by temporal alignment. These results establish Linear SVM as the strongest classical baseline for this task.

8.4.3 Random Forest

Random Forest achieved the highest accuracy among classical models (0.725) but a comparatively lower Macro-F1 score (0.297). This discrepancy indicates majority-class dominance rather than balanced event prediction. The model struggled to generalize across minority classes due to the sparsity and dimensionality of TF-IDF features, leading to class collapse despite high overall accuracy. This outcome underscores that accuracy alone is insufficient for evaluating imbalanced text classification tasks and highlights the importance of matching model architecture to feature representation.

8.4.4 Transformer-Based Model Performance (DistilBERT)

To assess if contextual embeddings improve event classification, a fine-tuned DistilBERT was trained on the same dataset. It achieved the highest accuracy (0.751) but a Macro-F1 of 0.270, slightly below the best classical baseline. Although transformer models are expected to outperform traditional methods, this is due to challenges posed by weak supervision, short input

length, and class imbalance. Despite this, DistilBERT showed stronger robustness to informal language, partial references, and implied context, especially for the majority and mid-frequency classes. Its performance suggests that contextual embeddings help model fan discourse, even when label noise limits gains on balanced metrics.

8.4.5 Error Analysis and Observations

Although DistilBERT achieved the highest accuracy, Macro-F1 scores remained moderate. Error analysis revealed limitations: many comments lack enough linguistic detail to identify specific events, and temporal alignment produces weak labels that don't always reflect comment intent. Severe class imbalance also suppresses performance on rare events. These challenges are data-driven, indicating that future improvements rely more on better alignment and richer context than on more complex models.

8.4.6 Interpretation and Implications

Supervised learning shows MLB in-game events can be accurately inferred from Reddit fan commentary, especially with transformer models. While TF-IDF provides solid baselines, contextual models excel with short, noisy text. These results confirm that NLP can detect events from fan discourse, though performance is limited by weak supervision and limited context. Future work should improve temporal alignment, include broader context, and expand datasets across games or platforms.

9 Deployment Plan and Conclusion

In a real-time production environment, a scheduled job continuously collects new Reddit comments, keeping inputs fresh. Each comment is cleaned, tokenized, lemmatized, and normalized, matching the steps used during model training for consistent predictions. The comment’s timestamp is aligned with the nearest MLB pitch timestamp to understand the game context in real time. A lightweight API hosts the fine-tuned DistilBERT model, which receives cleaned text and returns predicted event groups with confidence scores. System logs track predictions; if language patterns or prediction distributions shift, the system detects “model drift” and triggers retraining with updated data. Analysts can then view real-time event predictions, sentiment trends, topic activity, and comment volume via a live dashboard.