Clasificación de Reseñas para la Mejora en la Toma de Decisiones de Compras en Línea

Daniel Angel Arró Moreno, Pedro Pablo Alvarez Portellez, and Abel LLerena Domingo

Facultad de Matemática y Computación, Universidad de La Habana, Cuba

Abstract. Este reporte describe el desarrollo de un sistema de clasificación de reseñas para mejorar la toma de decisiones de compra en sitios web de compras en línea. Se resolvió el problema de la sobrecarga de información al proporcionar un sistema que organiza las reseñas con base en la similitud de usuarios, el análisis de sentimientos y un algoritmo personalizado de PageRank. Además, se presentan los antecedentes, el estado del arte, las soluciones implementadas, y las limitaciones del proyecto, junto con propuestas de mejora.

1 Introducción

El contenido de las reseñas generadas por los usuarios es crucial para que los compradores tomen decisiones informadas en sitios web de compras en línea. Sin embargo, el gran volumen de reseñas, especialmente en productos populares, dificulta la búsqueda de información relevante. Los métodos de clasificación ofrecidos por los sitios web suelen ser insuficientes, por lo que se propuso un sistema que aprovecha características avanzadas para mejorar la experiencia de los compradores.

2 Descripción del Tema

El tema principal de este proyecto es la mejora en la toma de decisiones de los compradores en línea mediante la clasificación inteligente de reseñas. Se reconoció la necesidad de un sistema más avanzado que permita a los usuarios encontrar rápidamente reseñas útiles y relevantes, lo cual es crucial para la satisfacción del cliente y para la toma de decisiones de compra más acertadas.

3 Antecedentes

Diversos estudios han señalado que las reseñas generadas por usuarios son un factor decisivo en el proceso de compra en línea. Sin embargo, el enfoque predominante en la presentación de estas reseñas sigue siendo limitado, lo que motiva la necesidad de investigar y desarrollar métodos más eficaces de clasificación. Los métodos tradicionales, como la clasificación por fecha o por calificación, no son

suficientes para extraer la información más relevante para los usuarios. Se han propuesto diversos enfoques que intentan mejorar la experiencia de usuario, incluyendo sistemas de recomendación basados en contenido, filtrado colaborativo y análisis de sentimientos, cada uno con sus propias limitaciones.

4 Estado del Arte

En el ámbito de la clasificación y recomendación de reseñas, los enfoques más comunes incluyen el uso de algoritmos de filtrado colaborativo, técnicas de minería de opiniones y análisis de sentimientos. El filtrado colaborativo, tanto basado en usuarios como en ítems, es una técnica ampliamente utilizada que recomienda productos o reseñas en función de patrones de comportamiento similares entre usuarios. Por otro lado, el análisis de sentimientos permite inferir la polaridad de las opiniones expresadas en las reseñas, lo cual ayuda a clasificar y priorizar contenido positivo o negativo.

5 Métodos Implementados

En este proyecto, se implementaron varios métodos y funciones para abordar el problema de la clasificación de reseñas:

5.1 Análisis de Sentimientos

Se utilizó la herramienta `SentimentIntensityAnalyzer` de NLTK para analizar el sentimiento de las reseñas. La función `analyze_sentiment` clasifica el sentimiento de cada reseña como positivo (1), negativo (-1) o neutro (0) basándose en un puntaje de polaridad.

Listing 1.1. Función de análisis de sentimientos

```
def analyze_sentiment(text: str) -> Literal[-1, 0, 1]:
    sia = SentimentIntensityAnalyzer()
    sentiment_score = sia.polarity_scores(text)

    if sentiment_score['compound'] >= 0.05:
        return 1
    elif sentiment_score['compound'] <= -0.05:
        return -1
    else:
        return 0
```

5.2 Creación del Grafo de Relaciones

La función `create_graph` construye un grafo bipartito donde los nodos representan usuarios y productos, y las aristas están ponderadas según el sentimiento y la calificación de la reseña.

Listing 1.2. Función para crear el grafo

```
    def create_graph(df):
        G = nx.Graph()

        # Crear nodos para usuarios y juegos
        for _, row in df.iterrows():
            G.add_node(row['reviewerID'], bipartite=0)
            G.add_node(row['asin'], bipartite=1)

            # Aniadir arista con peso basado en sentiment y overall
            weight = (row['sentiments'] + 1) / 2 * (row['overall'] / 5)
# Normalizar a [0, 1]
            G.add_edge(row['reviewerID'], row['asin'], weight=weight)

        return G
```

5.3 PageRank Personalizado

Se implementó un algoritmo de PageRank personalizado para ordenar los usuarios en función de su similitud en las revisiones de un producto específico.

Listing 1.3. Función de PageRank personalizado

```
    def personalized_pagerank(G, user_id, game_id):
        similar_users = get_similar_users(G, user_id, game_id)

        # Crear vector personalizado
        personalization = defaultdict(float)
        for user in similar_users:
            personalization[user] = 1.0

        # Ejecutar PageRank personalizado
        pagerank = nx.pagerank(G, personalization=personalization)
        return pagerank
```

5.4 Rango de Reseñas

La función `rank_reviews` utiliza el PageRank personalizado para calcular un puntaje combinado para cada reseña basado en la influencia del revisor, el sentimiento y la utilidad percibida.

Listing 1.4. Función para ordenar las reseñas

```
    def rank_reviews(df, user_id, game_id, top_n = 10):
        G = create_graph(df)
        pagerank = personalized_pagerank(G, user_id, game_id)
```

```
        # Filtrar reviews del juego especifico
        game_reviews = df[df['asin'] == game_id]

        # Calcular score para cada review
        scored_reviews = []
        for _, review in game_reviews.iterrows():
            reviewer_rank = pagerank.get(review['reviewerID'], 0)
            sentiment_score = (review['sentiments'] + 1) / 2
# Normalizar a [0, 1]
            helpful = [int(x) for x in review['helpful'][1:-1].split(', ')]
            helpfulness = helpful[0] / max(helpful[1], 1)

            score = (reviewer_rank + sentiment_score + helpfulness) / 3
            scored_reviews.append((score, review))

        # Ordenar reviews por score
        scored_reviews.sort(reverse=True, key=lambda x: x[0])

        return scored_reviews[:top_n]
```

6 Evaluación del Trabajo

La evaluación del sistema se realizó mediante un análisis cuantitativo y cualitativo de las métricas de desempeño, como la diversidad del contenido, la utilidad percibida y la cobertura temporal. Se observaron mejoras en la relevancia de las reseñas mostradas, especialmente en productos con un gran número de revisiones.

7 Métricas Implementadas

En esta sección se describen las métricas utilizadas en el proyecto para evaluar la calidad y relevancia de las reseñas de productos. Cada métrica está diseñada para capturar un aspecto particular de la reseña, con el fin de proporcionar un sistema de clasificación más preciso y útil para los usuarios.

7.1 Diversidad de Contenido (Content Diversity)

Descripción: Esta métrica mide la diversidad en el contenido de las reseñas utilizando la similitud del coseno entre los textos de las reseñas. La similitud del coseno es una medida de cuán similares son dos vectores en un espacio de alta dimensión, donde en este caso, los vectores representan las reseñas tras la transformación TF-IDF (Term Frequency-Inverse Document Frequency).

Fórmula:

$$\text{Content Diversity} = 1 - \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \cos(\vec{v_i}, \vec{v_j})$$

Donde:

- $\vec{v_i}$ y $\vec{v_j}$ son los vectores TF-IDF de las reseñas i y j.
- N es el número total de reseñas.
- $\cos(\vec{v_i}, \vec{v_j})$ es la similitud del coseno entre los vectores $\vec{v_i}$ y $\vec{v_j}$.

Propósito: Una alta diversidad en el contenido indica que las reseñas cubren diferentes aspectos y opiniones, lo que puede ser más útil para los compradores al tomar decisiones informadas.

7.2 Percepción de Utilidad (Perceived Helpfulness)

Descripción: La percepción de utilidad mide cuán útiles se perciben las reseñas basándose en la cantidad de votos positivos que han recibido en comparación con el número total de votos (positivos y negativos).

Fórmula:

$$\text{Perceived Helpfulness} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{helpful}_i}{\text{totalVotes}_i}$$

Donde:

- helpful_i es el número de votos útiles para la reseña i.
- totalVotes_i es el número total de votos (útiles + no útiles) para la reseña i.
- N es el número total de reseñas.

Propósito: Esta métrica ayuda a priorizar las reseñas que otros usuarios han encontrado útiles, lo que puede guiar mejor a futuros compradores.

7.3 Cobertura Temporal (Temporal Coverage)

Descripción: La cobertura temporal mide el rango de tiempo cubierto por las reseñas seleccionadas. Es importante para asegurar que las reseñas sean relevantes y estén distribuidas a lo largo del tiempo.

Fórmula:

$$\text{Temporal Coverage} = \frac{(\text{maxDate} - \text{minDate})}{\text{totalTimeSpan}}$$

Donde:

- maxDate y minDate son las fechas de la reseña más reciente y más antigua, respectivamente.
- totalTimeSpan es la duración total de tiempo en la que las reseñas podrían haber sido escritas.

Propósito: Garantiza que las reseñas no se concentren en un solo periodo, reflejando la evolución del producto y su recepción a lo largo del tiempo.

7.4 Cobertura de Aspectos (Aspect Coverage)

Descripción: La cobertura de aspectos evalúa cuán bien cubren las reseñas los diferentes aspectos o características de un producto. Los aspectos pueden incluir jugabilidad, gráficos, facilidad de instalación, etc.

Fórmula:

$$\text{Aspect Coverage} = \frac{1}{M} \sum_{j=1}^{M} \frac{\text{count}_j}{N}$$

Donde:

- M es el número total de aspectos definidos.
- count_j es el número de reseñas que cubren el aspecto j.
- N es el número total de reseñas.

Propósito: Asegura que se consideren todas las características importantes del producto, proporcionando una evaluación más equilibrada y completa.

7.5 Diversidad de Sentimientos (Sentiment Diversity)

Descripción: La diversidad de sentimientos mide la variación en los sentimientos expresados en las reseñas (positivos, negativos o neutros). Una alta diversidad indica que las reseñas reflejan una variedad de experiencias y opiniones.

Fórmula:

$$\text{Sentiment Diversity} = \frac{\text{number of unique sentiments}}{N}$$

Donde:

- N es el número total de reseñas.

Propósito: Permite identificar si las reseñas presentan una gama diversa de opiniones, lo que puede ser útil para entender los pros y los contras de un producto.

7.6 Cobertura de Calificación (Rating Coverage)

Descripción: La cobertura de calificación mide la variación en las calificaciones otorgadas por los usuarios. Similar a la diversidad de sentimientos, pero enfocada en las calificaciones numéricas.

Fórmula:

$$\text{Rating Coverage} = \frac{\text{maxRating} - \text{minRating}}{5}$$

Donde:

- maxRating y minRating son las calificaciones máxima y mínima, respectivamente.

Propósito: Una amplia cobertura de calificación puede ayudar a entender mejor cómo varían las opiniones sobre el producto entre diferentes usuarios.

7.7 Índice de Legibilidad (Readability Index)

Descripción: El índice de legibilidad evalúa qué tan fácil es leer las reseñas utilizando el índice de facilidad de lectura de Flesch.

Fórmula:

$$\text{Readability Index} = \frac{1}{N} \sum_{i=1}^{N} \text{Flesch Score}(i)$$

Donde:

- Flesch Score(i) es el puntaje de legibilidad Flesch para la reseña i.
- N es el número total de reseñas.

Propósito: Asegura que las reseñas sean accesibles y comprensibles para un público amplio, aumentando su utilidad.

8 Limitaciones y Propuestas de Mejora

A pesar de los resultados obtenidos, el sistema presenta limitaciones en cuanto a la escalabilidad y la precisión del análisis de sentimientos en reseñas con lenguaje ambiguo o sarcástico. Se propone la integración de técnicas de procesamiento de lenguaje natural más avanzadas, como modelos de lenguaje preentrenados, y la optimización del algoritmo de PageRank para manejar grandes volúmenes de datos. También se propone agregar otros elementos como el historial de compra y búsqueda del usuario para tomar mejores decisiones a la hora de filtrar los mejores comentarios.

9 Conclusión

El sistema desarrollado proporciona una mejora significativa en la clasificación de reseñas en sitios de compras en línea, facilitando la toma de decisiones de los usuarios. A pesar de las limitaciones, las propuestas de mejora ofrecen un camino claro para la evolución futura del sistema.

10 URL del Proyecto

`https://github.com/danielangelarro/Review-Classifier`