Silver Screen - Requirements, Vision, and Scope

Overview

Silver Screen is a web application which allows its users to gather and visualise popular movie related sentiments based on Twitter tweets. The derivation and categorization of these sentiments is based on metrics such as a tweet's positivity, neutrality, and negativity - all of which are calculated through the tweet's word composition and sentence structure. Silver Screen also offers visualisations of analyzed tweets, providing insight into their posting location, submission date, and other relevant parameters. Data is analysed using natural language processing, primarily in Python, and visualized through charts and an overall review score on the web front-end.

Motivation/Opportunity

We built this product because Twitter contains a vast, untapped supply of information which can be useful to a variety of industries, one of which being the movie industry. By consolidating this information, we can turn a relatively disorganized social network into a tool for understanding public reactions. This information can be used by the movie industry to fill gaps in their understanding about the success of a movie and by individuals looking to find new movies to watch.

Although there has been a large amount of research into using machine learning to process text based information and determine the sentiment in it, there are currently no existing products which present the results of these analysis techniques in a manner that is easy to access and understand. Websites such as IMDB and Rotten Tomatoes provide a way to view movie ratings generated through user contributions or professional reviews; however, our product provides a larger and more accurate database of reviews as it takes into account the opinions of people who are not inclined to take the time to write reviews on other websites.

Problem Statement

The Problem Of	Inaccurate movie reviews and disorganization of information contained within Twitter
Affects	Persons interested in the performance or popular opinion of movies
The impact of which is	People might be unsure as to how the general public feels about a movie
A successful solution would be	A software system that concisely and accurately portrays popular opinions on movies

Product Position Statement

For	Movie viewers, industry entities, and vendors (theatre-owners)
Who	Are looking for general public opinions on movies
Our System	Is a web service (all software)
That	Allows users to consolidate, quantify, and visualize popular opinions on movies through tweets found on Twitter
Unlike	Other websites like IMDB and Rotten Tomatoes which focus on reviews by professionals or others who are invested in the industry
Our Product	Is user friendly, accurate, free from selection bias, and fully available online

User Demographics

Movie-goers: The primary audience of our site are standard movie-goers who wish to use more accurate, crowd-sourced reviews to find well-regarded/popular movies. As movies appeal to an extremely wide range of the population, our site must attempt to target users of all ages and technical skill levels. These generally non-technical users expect seamless implementations similar to the ones implemented by other popular review sites (such as IMDB or Rotten Tomatoes), which are presented in a visually appealing, easy to follow format.

Industry Professionals: The movie industry can use Silver Screen in order to gauge public reaction to movie announcements, trailers, and the movies themselves. This information can augment focus groups, providing them with a much larger sample reaction to the information that they release. These users are generally more technically savvy than standard movie-goers, and are expecting the presentation of robust data which relates to their query.

Theatre Operators: Theatre Operators can use Silver Screen to identify which movies are trending at a given moment. They can then use this information to decide which movies to play in their future showings. These users are often somewhat-technically inclined, yet they still require a clean interface to ensure that they can easily access and interpret the information they will use to judge and select movies.

Feature List

- Web-based search interface
- Automatic analysis of the general impression of a movie on Twitter, including the generation of a review score
- A minimalistic front page which presents trending movies through a clean user interface
- Visualisation of Twitter data in user-friendly graphs and charts
- Presentation of summary information regarding all movies which Silver Screen has analyzed in the past
- Storage and display of historical records regarding people's' responses to a movie

Constraints

- Language: Given the scope of our project, and the limitations of the natural language toolkit we will rely on, it will only be possible to process tweets written in English. Of course, this may affect the accuracy or scope of our results.
- Accuracy of Language Toolkits: While the tools used in sentiment analysis allow for a correct determination of sentiment in a majority of cases, they are far from perfect. As such, the review scores we generate are not completely accurate and are resultantly "fuzzed" to avoid implying pinpoint accuracy.
- Available data: It is quite likely that people will not have tweeted about certain movies very much. Therefore, for certain search queries, the system may not receive enough information from Twitter to make accurate statements about people's views on that movie. Moreover, Twitter only stores tweets for up to one week, so our application is not able to graph sentiment scores for older tweets which we have not accessed in the past.
- Base Operating Environment: We have chosen to develop our application using Python's Django framework, allowing us to focus on natural language algorithm design and data extraction over development specifically for the web.
- Database: Our choice of database software was directly influenced by our web host. As we opted to use Heroku (a decision detailed in the design document), we were required to use PostgreSQL.

Scope and Limitations

- Silver Screen is targeted towards movie-related sentiments, and will not return sentiment scores for other mediums (i.e. TV shows, books, etc.)
- Due to the difficulties stemming from translation and parsing, analysis is restricted to English tweets only.
- Silver Screen does not allow for users to create accounts in order to avoid a large amount of login system-based overhead

- When analyzing movies which are part of major franchises, review scores are generally based on the franchise, not the individual movie. One example would be the 'Mission Impossible' franchise - a specific movie in this franchise (such as 'Mission Impossible: Ghost Protocol') will likely have little to no tweets which are exact matches, and hence we are required to simplify the query (i.e. 'Mission Impossible') to find more results at the cost of relevancy.

Assumptions and Dependencies

This project relies heavily on Tweets and thus makes extensive use of the Twitter API. We also work under the assumption that there are a statistically significant number of meaningful Tweets about most movies the users of the application would be likely to search for. We also assume that we are able to request and process enough tweets in order to keep sentiment scores for movies updated in a timely manner.

Given that we will be breaking down and analyzing Tweets using Python's Natural Language Toolkit, we will be relying on its abilities, and our own, to perform meaningful and reasonably accurate analysis of subjective information.

Use Cases

Use Case 1: User requests general sentiment information about a specific movie

Stakeholders and Interests	The user is the stakeholder here. He/she is looking for accurate popular perspective on a specific movie.
Preconditions	The user has loaded the webpage and entered his/her query into the search box, no login required.
Postconditions	The user interface will display the sentiment data in a user-friendly manner.
Main Success Scenario	1. The user presses search (via mouse or keyboard) a. If required, a loading screen is displayed to the user 2. The user is shown basic information about the movie (title, release date, etc.), an overall sentiment score, controversy rating, review scores from other review sites (IMDB and Rotten Tomatoes), and data visualizations in the form of dynamically generated graphs 3. The user can switch between

	graphs/visualizations using a tab-based interface 4. If desired, the user can expand a drawer to see a curated selection of tweets (randomly chosen or based on the number of retweets)
Extensions and Alternative Flows	1. There are not enough tweets for the movie a. If possible, the system will attempt to simplify the search i. If successful, display sentiment data related to the simplified search, with the user being warned through an unobtrusive notification ii. If unsuccessful, return the user to the home page and display an error message 2. An API connection is broken, a movie with the provided title (or similar) cannot be found, or a server exception occurs a. Return the user to the home page and display an error message
Open Issues	What information should be displayed on the loading screen (if applicable)?

Use Case 2: User wishes to discover a trending movie and obtain general sentiment information about it

Stakeholders and Interests	The user is the stakeholder here. He/she is looking for information on popular/trending movies.
Preconditions	The user has loaded the webpage, no login required, and there exists at least one movie that has surpassed the preset number of views to consider it a “trending candidate”.
Postconditions	The user interface will suggest a movie and display its sentiment data in a user-friendly manner.
Main Success Scenario	1. Upon entering the site, the user is presented with the name of the movie which has been searched for the most in the past 24 hours (assuming it is a “trending candidate”) in a small alert box

	2. If the user wishes to view the movie's corresponding sentiment page, they may click the alert box to be directed to the corresponding sentiment page. 3. *The flow then follows use case #1*
Extensions and Alternative Flows	1. There are no movies which are "trending candidates" a. The trending alert box is not displayed to the user
Open Issues	How often should we wipe a movie's trending view count to ensure that the movie we suggest is actually representative of what is currently popular?

Use Case 3: User wishes to discover a well-regarded movie and obtain general sentiment information about it

Stakeholders and Interests	The user is the stakeholder here. He/she is looking for information on a random, well-regarded movie.
Preconditions	The user has loaded the webpage, no login required.
Postconditions	The user interface will suggest a movie from IMDB's top 250 movies list and display its sentiment data in a user-friendly manner.
Main Success Scenario	1. Upon entering the site, the user is informed via the search box's default text that they can leave the search box blank to retrieve information about a highly-regarded film 2. The user leaves the search box blank and presses search 3. A random movie is retrieved from IMDB's top 250 movies list, and the flow subsequently follows use case #1
Extensions and Alternative Flows	1. There are no relevant tweets for the randomly selected movie a. A different movie from the top 250 list is randomly selected and displayed (the user is not informed about this issue)

Open Issues	In the future, should we allow the user to randomly select from a list of poorly-received movies?

Use Case 4: User wishes to see a summary of all sentiment scores generated to date

Stakeholders and Interests	The user is the stakeholder. He/she wishes to view information about all of the movies that have been searched to date.
Preconditions	The user has loaded the webpage, no login required.
Postconditions	The user interface will display a summary of the sentiment data for all of the movies in the database in a user-friendly manner
Main Success Scenario	1. The user navigates to the summary page through a navigation link (found on every page) 2. Statistical information regarding the sentiment data collected by Silver Screen (such as view counts, highest and lowest ranked movies, etc) is displayed to the user through text and other visualizations (such as graphs) 3. If the specified visualization references a certain movie, the user may click on its title to be redirected to its corresponding sentiment page
Extensions and Alternative Flows	1. Not enough searches have been performed to provide meaningful data (*Note: this will only occur if our movie database is wiped/reset, and should normally not occur)* a. No data is displayed, and the user is alerted that they should return to this page later

Non-Functional Requirements

Performance Requirements

Due to the volatile nature of Twitter, tweets have to be pulled regularly to ensure that review scores remain current. It is important, however, to weigh this requirement against the issue of load times within Silver Screen - pulling, processing, and presenting sentiment for a new movie often takes a few seconds to execute, even with multithreaded API requests. As a result, our system attempts to rely on data which the user does not have to wait for (i.e. showing previously pulled tweets). We unfortunately cannot avoid wait times 100% of the time, however, as if a user requests information about a movie that has not been processed previously we still must serve the request.

Safety Requirements

One possible safety issue is the use of people's tweets and their misinterpretation by our software. As we will only be using public tweets which are, as specified in Twitter's terms and agreements, public property, we believe this issue should be fairly contained. The user may be able to view a short list of movie-related tweets with their corresponding authors; however, their weighted contribution to the ranking of a movie will not be visible. A notice is also provided on the about page, clarifying our approach and limitations of our methodology.

Security Requirements

Our system will guard against users submitting many queries within a small window of time (maliciously or otherwise). This safeguarding is critical in preserving the number of API calls we are able to perform, as our API licence only supports a limited number of twitter requests per 15 minutes. If we reach our quota, users will only be able to access previously processed movies.

Software Quality Attributes

We are not considering adding an API for our software so it will not be adaptable by others to perform their own analysis using our algorithm. However, if we wish we can easily expand the Twitter analysis to analysis of other sites after project completion and presentation.

Our software is readily available on a website and it can be accessed on a PC or capable mobile device. Information should be presented to users in an attractive and simple, yet informative manner.

We have a scale of correctness which correlates with the amount of feedback we obtain about a movie. The more opinions we get about a movie, the better we can form a correct representation of the relevant public opinion.

We have a few dependencies that will need to remain intact throughout the software lifecycle - mainly the Twitter API, Django and Python versions. In order to keep our software maintainable

we will need to make sure that these versions are tracked and are consistent throughout development and launch.

Design Changes and Rationale

Throughout the development of Silver Screen, we have modified this document to represent changes to this project's requirements. The following list of changes is included below.

Use case removal: User Requests Historical Data
The preliminary requirements documentation outlined a use case where users were able to request sentiment about a movie over a large time period (from the movie's release date to the present). However, after learning that Twitter only stores data up to one week we have decided to remove this requirement as it is no longer realistic. Given that we store the sentiment data which we collect through user requests over time the historical aspect of Silver Screen will be achievable for newer movies, but not to the degree we originally intended.

Use case additions: Trending/Well Regarded Movies
While the original requirements documentation alluded to presenting the user with popular/trending movies, there was no specific discussion on how these movies would be presented to the user. We have therefore elaborated on these presentations through use cases #2 and #3.

Use case addition: User Requests Database Summary
The preliminary use case documentation outlined a scenario in which the user could request comparisons of movies based on different search parameters. We found, however, that implementing a method to search for parameters which are not movie titles turned out to be prohibitively complicated due to API limitations. We have instead decided to revise this use case to allow the user to view an overall summary of all movies which we have information on - detailed in use case #4.

Review Score Update Period
In the original version of this document, we noted that we would be pulling tweets and updating sentiment scores through an automatic job, likely every 15 minutes. In developing our application, however, we have found that approach unfeasible due to the amount of unnecessary processing required (we would be constantly updating our database, even if we had no visitors). Rather than having an automatic job, we have instead decided to pull and process tweets upon a user's request.

Update of Security Requirements
This document originally detailed the requirement of sanitizing data to avoid SQL injection and other similar attacks. We have noted, however, that the Django platform properly escapes queries for us, and hence we are safeguarded against such attacks. We have replaced this section with further discussion regarding issues resulting from users issuing too many requests.

Lessening of Geographical Focus

While our original vision of Silver Screen aimed place a focus on correlating sentiment to geographical location, we have found that Twitter users' self-reported locations are often undecipherable or incorrect. For instance, some Twitter users use non-geographical/joke locations (such as "Heaven" or "Mars"), and some users use location names which can be mapped to multiple areas (such as "London" or "Vancouver"). As a result, we have lessened our focus on geographical analysis and cancelled the map plotting feature.