TROPEFY | Interactive movie recommendation system using movie tropes

TEAM 63

Venkatasatyanarayana kamisetti (vkamisetti3@gatech.edu) | Narendra Gadidasu | Ritu Parathody | Ramesh Arvind | Girish Rao | Kunal Singh

Introduction

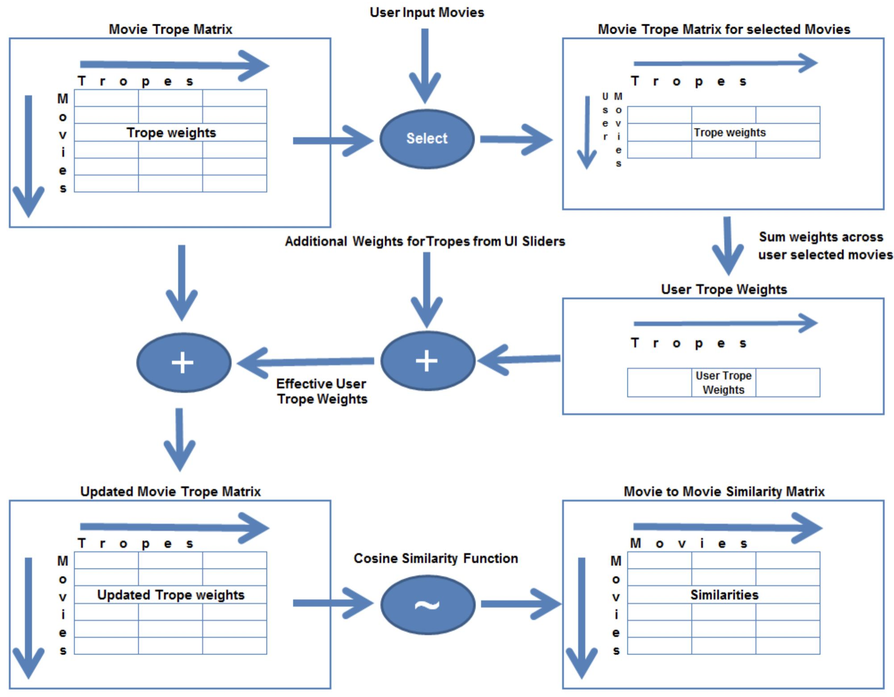
Have you ever felt that the movie recommendations shown to you are not very transparent and engaging? Have you ever felt the need have control over the tuning knobs? If the answers is yes, that's what **Tropefy** offers with the intention of transparent and story based recommendations

Tropefy merges robust **recommendation algorithm** backed by **movie tropes** with an **interactive visualisation**, to help users search for movies.

Most of the existing recommendation engines present results as sorted lists. As a result, user can't understand the structure of the recommendations.

Algorithm

We used a **modified collaborative filtering** with starting point as matrix with movies as rows and tropes as column. Each of the elements of the matrix, will be either 0, if a trope is not present in movie or it would be the Trope Weight, if it is present.



An input of one or more movies is taken from the user and for each= trope, User Trope Weight (UTW) is calculated as the Trope Weight (TW) times the number of user selected movies in which the trope is Present.

Similarity between movies i and j is defined as: $\sum_{t}^{T} w_{it} w_{jt}$

where t = 1,2..T are various tropes and represent the weight of trope 'y' in movie 'x'. The Effective Score (ES) for each movie would be the sum of normalized values of Final Similarity Score (FSS), vote_average and popularity extracted from TMDB.

Tropes

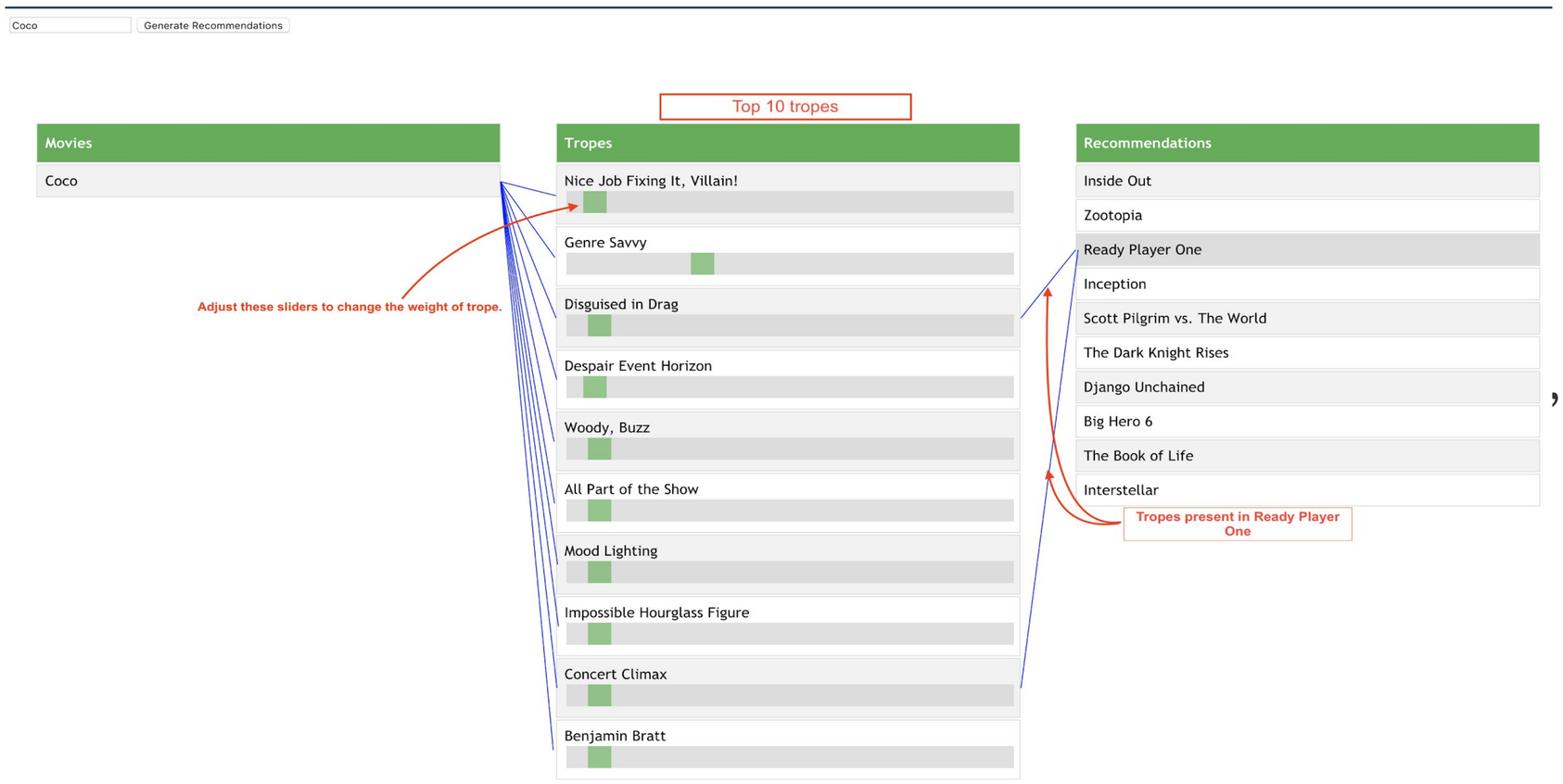
Tropes are recurrent themes and narrative frames that appear throughout literature, news, and popular media. Movie tropes are wiki that helps the viewers to identify and document the similar tropes in the media and literature.

Data

We scraped https://tvtropes.org/ and movies released between the years **2010 and 2019**. We also collected all the tropes and corresponding trope details collated from all the selected movies. To get additional data about the movies we also collected it form TMDB website.

- Total number of movies: 2261
- Total number of tropes: 126174
- Size of the movies data on disk: 3.7M
- Size of trope details data on disk: 31M

Visualization



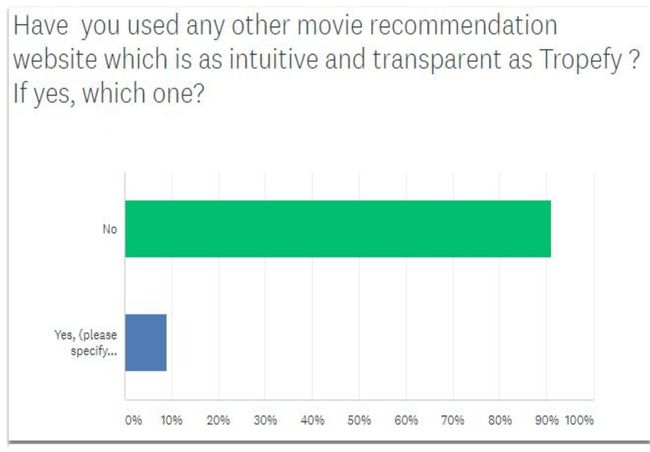
Proposed approach was more focused on representing the relationships between the recommend movies and allows user to give feedback by controlling the recommendations itself by changing weights in the list. This approach doesn't capture as to why the user disliked the particular recommendation and gives little feedback to the algorithm to refine new recommendations

Evaluation

In this application, we are trying solve the case of recommendations where there is no ground truth data.. Evaluation in such a situation can't be done with typical quantitative evaluation metrics. So we hosted our application and distributed it to a sample of users to get feedback via survey

- 90.9% of the users found Tropefy more intuitive and transparent than any other system
- 36.6% of the user found recommendation parameter more transparent in Tropefy than TasteDive.
- 45.45% of the users found the system somewhat intuitive to their taste in movies.

Our method of evaluation is very different from the existing methods which generally rely on existing user features like demographics, similar user ratings, overall movie ratings, etc. Since our platform doesn't support a inherent feedback mechanism from the user at this stage, it's difficult to quantify the experience. Also, the basic idea of our platform is to provide a near to personalised experience and hence, we didn't used overall movie ratings from website like IMDB or TMDB for evaluation.



We hosted our system at http://rameshharvind.pythonanywhere.com/ui