Age and Context Sensitive Entertainment Recommendation System for Families

Abhinn Yadav	Aman Kumar	Dhruv Mishra	Hunar Kaur	Neelabh Kumar Srivastava	Sadhvi Bhan
2020013	2020279	2020296	2020303	2020317	2020325

Literature Review

In this section, we first discuss what we intend to derive from a particular paper, that is how the paper may be relevant to our project and then proceed to explain the findings in the literature review.

[1] User reviews provide relevant information about user sentiments.This can be helpful in collaborative filtering as discussed in project proposal, reference [3]. User reviews is a form of unstructured dataset. Hence, we require methods for information retrieval from the review.

This paper compares two feature extraction techniques of N-Gram and TF-IDF on IMDB movie reviews and Amazon Alexa reviews dataset. For both the models, user review is first made punctuation and stop-words free. It is then tokenized. For n-gram, a bigram(n = 2) has been made. Six different classifiers SVM, Decision Tree, Multinomial Bayes etc have been used. Finally, they all have been compared on the basis of accuracy, recall, precision score etc. It is found that for n-gram approach, Logistic regression was the best classifier while in the case of TF-IDF, SVM and Random Forest were the better classifiers. The maximum accuracy, precision, recall (93.81%), and F1-score (91.99%) value was obtained by TF-IDF method using a Random Forest classifier.

[2] Our spelling correction and autocorrect will not only be context-sensitive but will also return age appropriate spelling suggestions. Like ‘viole’ should be corrected to ‘violet’ and not ‘violent’ if the user is not an adult. From this paper, we intend to draw effective methods to penalize age-inappropriate suggestions.

The following paper talks about implementation of an unsupervised context sensitive spelling correction method which generates replacement candidates and then ranks them according to their semantic fit. They first generate replacement candidates with an edit distance of 2 from a reference lexicon. They use a cosine weighted similarity between vector representation of a candidate and the vector representations of the misspelling context. If the candidate is an OOV(out of vocabulary), then a penalty is imposed.

[3] The authors talk about how query auto completion provides poor predictions of the user's query when the input prefix is very small. They show how context like the user's recent queries can improve the prediction considerably. They propose a context-sensitive query auto completion algorithm, NearestCompletion, which outputs the completions of the user's input that are most similar to the context queries. To measure similarity, they represent queries and contexts as high-dimensional term-weighted vectors and resort to cosine similarity. The mapping from queries to vectors is done through a new technique that they introduce, which expands a query by traversing the query recommendation tree rooted at the query.

When the context is irrelevant NearestCompletion's MRR is essentially zero. To fix this problem, they propose HybridCompletion, which is a hybrid of NearestCompletion with MostPopularCompletion. HybridCompletion is shown to dominate both NearestCompletion and MostPopularCompletion, achieving a total improvement of 31.5% in MRR relative to MostPopularCompletion.

Updated Problem Formulation:
We aimed to design a full-fledged movie recommender application that recommends movies based on the age of the users. The initially decided features of our recommender system were a search bar with age context-sensitive spell check, a search filter in a movie review, and a ranking of movie recommendations. However, researching more, we found out that using filtering techniques and then implementing a ranking model will lead to redundancy; hence we decided not to implement it.
We implemented the search bar feature by implementing boolean and phrase query searching using unigram inverted index and positional inverted indexing techniques respectively. The age-sensitive filter for the same is yet to be added. For recommending similar movies, we have implemented a content-based filtering technique that recommends movies based on the user's likes. To implement content-based filtering, we converted the description of the user's preferred movie and other movies in the database into TF-IDF vectors and then compared the vectors using cosine similarity. Higher similarity constant means that the movies are similar. The age-sensitive filter has also been added.

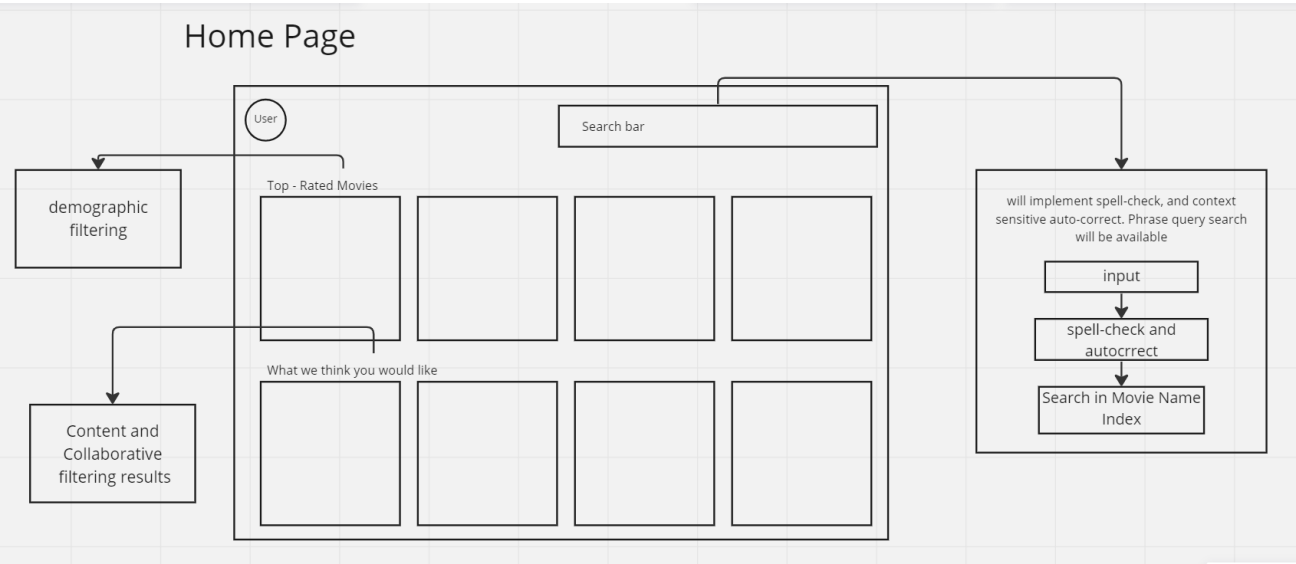
Updates to previous problem formulation:

1. No need for separate ranking models
   We initially researched various ranking models and algorithms. For example, RankingModel() is a model which we had considered using. However, we found that we may not require it. We will be finding cosine similarity between movie descriptions and use it directly to rank the movies.

2. Incorporation of genre and description in search bar: While implementing the search bar feature we found out that instead of just searching on the basis of title, more relevant and vast suggestions were made when searching was done on the basis of title, genre and description.

3. Use of Google USE -
   Universal Sentence Encoder (USE) is a pre-trained Encoder which can be used in a variety of tasks (sentimental analysis, classification and so on). We aim to use this encoder to make our information retrieval more inclusive and context aware. Using this, we can map similar phrases and make our system better. For example, ‘wizard’ and ‘magic’ are words which may appear in similar movies, so if a movie description contains the word ‘wizard’ then another movie with ‘magic’ in its description will be recommended.

Prototype Design:
We designed some low fidelity prototypes for the frontend of our application. This is the Home Page of our application.



Baseline Results:

1. Query Processing using Boolean Search - Search Bar
   This is our prototype for the search bar. Whenever a user enters a word, probably looking for a particular movie, we want to display all movie titles in which the query word/phrase occurs in the movie name or description. Sometimes, a user may not recall the complete movie name, and may rather enter a description for the movie, hence description has also been considered.
   In this, for each movie we first tokenized the movie name, description and genre. Using these, a unique cumulative token set was generated for each movie. Now whenever the user inputs a word/phrase query, it is converted to a boolean query and movie names corresponding to matching sets are returned.

```
Original Input Query: jurassic
Number of Documents: 4
Movie Names:
LEGO Jurassic World: The Indominus Escape
Jurassic World Camp Cretaceous
LEGO Jurassic World: Legend of Isla Nublar
LEGO Jurassic World: Secret Exhibit
```

search results for keyword "jurassic"

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Original Keywords: jurassic world
Generated Query: jurassic or world
Number of Documents: 764
Movie Names:
LEGO Jurassic World: The Indominus Escape
Jurassic World Camp Cretaceous
LEGO Jurassic World: Legend of Isla Nublar
LEGO Jurassic World: Secret Exhibit
The Most Assassinated Woman in the World
Marc Maron: Too Real
The World We Make
Christiane Amanpour: Sex & Love Around the World
We Bare Bears
Animal World
Skylines
The Grandmaster
Birders
USS Indianapolis: Men of Courage
20 Feet From Stardom
SMOSH: The Movie
Mad World
Clive Davis: The Soundtrack of Our Lives
Scott Pilgrim vs. the World
...
```

Success

search result for keyword "jurassic world"

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Original Keywords: action adventure fantasy
Generated Query: action or adventure or fantasy
Number of Documents: 2564
Movie Names:
Automata
Good People
Kidnapping Mr. Heineken
Moonwalkers
Next Gen
Black Panther
Animal World
Hell and Back
Black Panther
In the Shadow of the Moon
Norm of the North: King Sized Adventure
Good People
Kidnapping Mr. Heineken
Moonwalkers
Next Gen
Archibald's Next Big Thing
Hell and Back
Black Panther
Sturgill Simpson Presents Sound & Fury
...
Top Grier
Used Goods
Viking Destiny
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Search results for genre

2. Movie Recommender:
   We implemented content-based filtering for recommending movies. The idea behind content filtering is that, given a movie, the program must find and recommend a similar movie to the user. To compare similarities between the plots of the movies, we planned to use the movie description column of our dataset.
   We first converted our string description into a vector. We did this by calculating the Term Frequency-Inverse Document Frequency (TF-IDF) vectors of the description of the movies. We did this for all the movies in our dataset.
   However, before calculating the TDIF matrix, we enforced a strict age rating filter on the dataset and filtered out all inappropriate entries. We did this to ensure that the content is family-friendly and that no viewer is exposed to potentially inappropriate content.
   For quantifying similarity, we compared different similarity metrics(Cosine, Pearson, and Euclidean) and found relatively indifferent rankings across the three. We finally settled for Cosine Similarity as it is independent of

magnitude and is much faster to calculate than the other two metrics without significantly affecting our final results.

```
Initializing Recommender...
Recommender Initialized

Now Cleaning The Dataset...
Dataset Cleaned...

Now Ready To Recommend...
Enter the Total Number of Users:2
Enter Your Age:15
Enter Your Movie Preference:#realityhigh
Enter Your Age:16
Enter Your Movie Preference:Care of Kancharapalem
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8800 Movies and TV Shows...
Here's What We Think You'll Like:
        1 Mr. Young
        2 Big Stone Gap
        3 A Very Special Love
        4 Kocan Kadar Konus
        5 Just Friends
```

Content Based Filtering for a groups of teenagers

```
Now Ready To Recommend...
Enter the Total Number of Users:3
Enter Your Age:19
Enter Your Movie Preference:Shutter
Enter Your Age:29
Enter Your Movie Preference:Phobia 2
Enter Your Age:23
Enter Your Movie Preference:Death of Me
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows.
Here's What We Think You'll Like:
        1 Unsolved Mysteries
        2 13 Reasons Why
        3 Fear Files... Har Mod Pe Darr
        4 Malevolent
        5 The Unborn Child
```

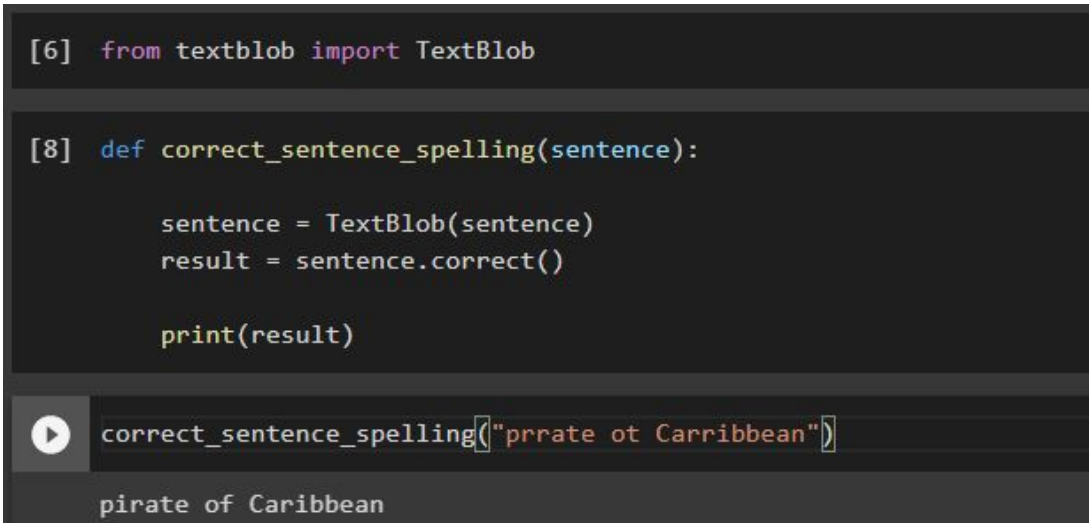
Content Based Filtering for an average urban household

3. Spell Check and Correct

   We have used TextBlob Library to implement context sensitive spelling correction. This will later be merged with the search bar to auto-correct movie names in case the user enters a wrong spelling. This will prevent the search bar from returning blank results.

   The current version demonstrates that while this library works, there are still some loopholes. For example, if the user enters: Pirates of Caribean (incorrect spelling of Caribbean) then this library fails as Caribbean is a proper noun. For this, we plan to incorporate the proper nouns present in movie names of our dataset.

```
[6]  from textblob import TextBlob

[8]  def correct_sentence_spelling(sentence):

         sentence = TextBlob(sentence)
         result = sentence.correct()

         print(result)

     correct_sentence_spelling("A sentencee tto demonstratt spell carraction !")

     A sentence to demonstrate spell correction !
```



References

[1] A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset Mamata Das, Selvakumar Kamalanathan and P.J.A. Alphonse

[2] Unsupervised Context-Sensitive Spelling Correction of Clinical Free-Text with Word and Character N-Gram Embeddings Pieter Fivez, Simon Suster

[3] Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In Proceedings of the 20th international conference on World wide web (WWW '11). Association for Computing Machinery, New York, NY, USA, 107–116. https://doi.org/10.1145/1963405.1963424