A DATA-DRIVEN MOBILE COMMUNITY APPLICATION FOR BOOK RECOMMENDATION AND PERSONALIZATION USING AI AND MACHINE LEARNING

Lulu Zha

Crean Lutheran High School, Irvine CA, USA

ABSTRACT

Knowing a movie or a book fits your flavor without finishing the whole film or the book? Although there are many ways to find a summary of a film or a book, having an app that generates needed information according to the genre will make things much more manageable. This paper develops a mobile app named Book and Movie Search that uses API or the online database to generalize data such as authors, plots, overview, and more with a few clicks. The results show that within seconds, a list of information will show according to movies and books, and a qualified way to find information using the Book And Movie Search app. For example, if one decided to buy a book named Flipped and did not have time to finish the whole book, he can enter the name on the app. It will generate a book summary that quickly gives him more information about it and help him decide whether he wishes to make the purchase.

KEYWORDS

technology, movie, book, search engine.

1. INTRODUCTION

Books and Movie Search app using the online database or the API and coded it with flutter in android studio. This app contains three split screens as well as a welcome page. The first split-screen has five pictures that list several popular books; the second split-screen is the book search, allowing people to search for books. There are four recommended books, including Animal Farm, Harry Potter, Flipped and lemons, and the author's publication date. At the very top of the screen, the search engine will gather information such as author, overview, publication date, and country of the publication as the user enters the book's name. The movie search engines contain similar content on the last and third split-screen. It has four recommended books, including Spiderman-no way home, Avengers, and Encanto, along with the director and the year it was published. As the user enters the movie's name, the director, plot, overview, and year of publication will show up. This app is designed for people to use before watching or reading a book. It will be easy for people to determine whether they wish to read/manage the content with a summary. For future books and movies, the app will automatically renew itself. It will update the new information and update the database used. The reason for me to build an app that generates books and movies summarizes that I think it will save a lot of time. To have a summary quickly determines whether someone is interested in the whole story or not.

Once I wanted to read Animal Farm, I decided to search whether it was fun. I got too much information about this book, specific to every character. Or I get a lot of related but not helpful

David C. Wyld et al. (Eds): ICAIT, CBIoT, WiMo, CRYPIS, ICDIPV, CAIML, NLCA - 2022
pp. 221-230, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121218

information such as the biography of George Orwell. After that, I found searching things on google, especially summaries of books, very time-consuming, and I wanted to create something easier to use. I hope this app will generate the most helpful information and provide a simple overview of the whole story, and they can choose whether to read the book or not.

One related method of searching this topic was Douban, an app that generates the most popular movies and books through commentary with the audience. Douban provides discussion groups and servers based on books and films, and through that, there will be a detailed, editable everyone information list that the users can view. However, no "official" summary exists for any books and movies; it is more like a discussion on a particular book/movie. Although they proposed to make an environment that allows movie and book lovers to share their thoughts, the accuracy of the content was limited. Douban, as users enter the names of books and films, will show descriptions of the movies and books. Although it allows for a longer description of a specific film, it will enable all users to edit the information. Due to the edibility of the information, anyone who has access to the internet can change the content, which will later result in the inaccuracy of the movie's content. For example, one wishes to have a brief overview of Charlie and the Chocolate Factory and use Douban. He will get a detailed result based on the movie, but with the risk that others might edit the content he is reading, he is likely to read something biased and might contain incorrect information. For example, if a viewer did not like Johnny Depp, the actor in the first place, he might watch the movie more on a stricter side and focus on whether Jonny Depp acted well. If he believes the opposite, he might go onto Douban and edit the content information based on how he felt. In that case, the content that people read could be biased, and that with possible false information will not be accurate and should not be used as information for either books or movies.

For this paper, I aim to explain how the app I created using API and flutter can generate information about books and movies and provide a summary of films and books in a few clicks. Douban inspires my method, and I liked how it allows people to create a discussion group to share thoughts on books and movies. However, it does not contain information about the movie or book in most cases. In the future, I want to create an app that gathers data to provide a quick overview of either books or films, which will help people decide what to read and watch quickly. There are some excellent methods for this app. First, on the first split-screen, there are four available book recommendations, and on a separate page, there are general recommendations for movies. Second, an extensive search engine allows people to input either books or films, and within a few clicks, a list of information will be shown, such as the author, the year of publication, and the summary of a book. For movies, there will be information such as the producers, authors, and an overview of the film.

As previously stated, the app's usefulness is based on the results shown for both books and movies. The results should contain various things such as authors, summary, the producers, and more. I needed to make sure the data I was getting was accurate to find information about the overview of books and movies; I could use the app Books and Movie Search. In two application scenarios, I demonstrated how the combination of accuracy and the techniques show that this app's results are accurate and can be used. In the first case study on the evolution of the consistency of results, I randomly picked twenty movies. Flip was the movie I chose to test the app, and I entered the film into the search engine. If the app gives the same result every time I search for the same thing, it shows consistent results. I tested five times and got the same feedback from the app, including the language, producer, and overall summary of the movie.

The second thing I tested was thewed accuracy of the results I got; since I was using an API, I needed to make sure that the results I was getting were correct or matched what was on the internet the most times. I again used the movie Flipped and checked the information I got from

google results; I needed to ensure that the producer, the language, and the publish time were accurate.

Based on the two tests, I concluded that the feedback was accurate and consistent.

This research paper follows the structure of an introduction paragraph, whether I showed what the app was about and how I managed to build it out. Section two was about the challenges I faced building this app named Books and Movie Search. Section three was about how I overcame the challenged I got previously stated in section two. Section four was about the details that shoed of the experiment I did as I tested and perfected mistakes. Section five was about related works that inspired me to change possible errors and learn from similar things that already existed in the market. Lastly, Section six was about the conclusion and future things I might do to improve my app.

2. CHALLENGES

Writing a research paper is a challenge, as picking a topic to write is very hard. There are many things that I wanted to include in my research paper, and I was trying to pick out a title that provides for everything. I tried to name my research paper something like how to connect the results of an AI app better and how it benefits the community, but I thought it would be too broad. After several tries, I still found it hard to decide what topic to dive into. There are many things to consider, and it took me a long time to finally start writing about the app I created, the Books and Movie Search. This app is not only a search engine. There are ideas behind it I tried to include, and I wish to develop them in the future further.

Organizing my work was also challenging, and it was hard to recall the steps I did as I did this app, and I was trying to combine things nicely. There are server things in the app, and I was trying to decide what came first and what came after, and I found it very hard to organize the language as I decided what to do. I looked through the codes, and the previously made slides helped a little. I tried to recall the steps as I wrote the research paper, but sometimes I was clueless about how I came up with some ideas, making me choose the topic. Overall, it was a challenge because it was hard for me to remember things nicely, and I could not, for the time, recall what I wanted to do as I designed the parts of the app.

The third challenge I faced was that although there are similar methods on the market, it was hard to source and compare with different things because not everyone had written a research paper on it. As I browsed through Google Scholar, I found a lot of related works but not the same things, and It was challenging for me to learn something from similar methods that there are, in fact, very few research papers that explained what they did on the plans. As I put in a book or movie search, many things showed up, but the reports discussed the pros and cons of the search engine itself instead of developing one on themselves. I had to find others who helped me get to the development of search engines, but if I put search engines, not all search engines specialized books and movies, and it was a challenge to find sources.

3. METHODOLOGY/SOLUTION

The basic idea of the Books and Movie Search was simple; the idea behind it was to find a way to use API and gather information for books and movies. This information will help users find the essential information for them and help them choose the proper film to watch or the right book to read in a few seconds by providing the summary. Since API is an online database containing such

information, the app will process and limit the search resource to only a few things, including the author, year of publication, summary, book and movie image, and more.

This app's structure connects the database with four other pages, including the splash screen, the search screen, the book detail page, and the movie detail page. As users open the mobile app books and movie search, there will first be a welcome page that will allow users to enter their names. To get started, there is a general page with four book recommendations on the page, and on the other page, there are another four movie recommendations on the top of the screen. These four recommendations provide the most popular or classic ones, allowing users to know what might appear in the search. In addition, I picked the movies and the books with friendly front pages so that it is cheerful to see as people enter the page. There are two separate tabs on the first screen: the bottom, the movie engine, and the booking engine. The first screen is a book search engine where people can enter the book's name to get a list of information, including the author and the book's image. It took me some time to decide whether I should contain the movie poster or the idea of the book; I only wished to have the essential pieces of information to limit the reading time and help people decide whether they are interested in the content. However, I found that a lot of times, it was the poster of the movie that hooked my interests, and the same thing applied to books. Therefore, I came to the idea of adding images to the app. Although it meant more coding to do, I think it worked out nicely. Whenever I was interested in some book name, it told me that for the time, I might also be hooked on the front page of the book. In addition, images would make the app pretty and pleasant; for that, I used to love comic books, and flipping through the books and scanning through the images had always been my favorite. Also, throughout this research paper, I always emphasized the importance of using the summary. However, having images along with the text would help things work out smoothly. As I was testing the app by entering random names for the booking engine, there were sometimes interesting front pages that interested me.

On the next page is the movie search screen. There will be a list of recommendations on the top, and if people enter the movie name from the movie search engine, there will also be four recommendations on the top of the movie search. Suppose people enter the name in the movie search engine. In that case, there will also be a list of information from the screen, including the producer, the language, the year of production, and an overview of the movie.

The steps are simple, the database connects everything, and there will be a separate screen that combines the information. From the app, there are only two main things to search for; one is for movies and one for books.

The app's components were the two split screens, the intro page, and the search screens. The app was supposed to make things easier; for that, it allows people to read the essential information, and it gathers stuff in a short amount of time.

Overall, the app's structure was to connect the database with two different search engines. By gathering and limiting the search result, the app will help its users find the right movies and books, provide background information, and help them decide what books and movies interest them in a short amount of time.

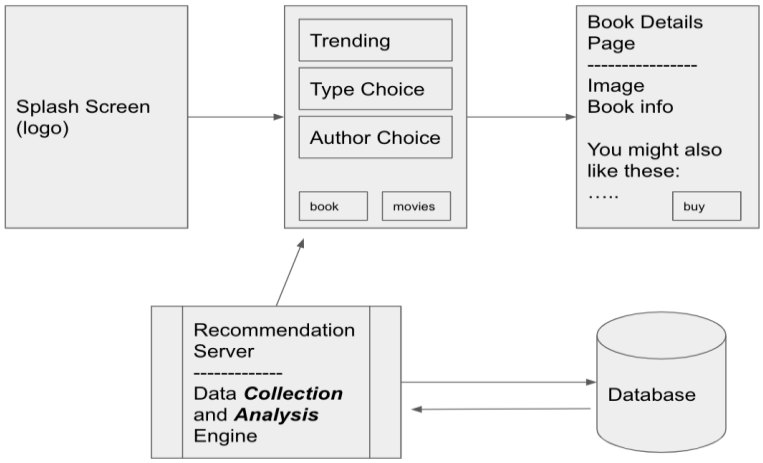

Figure 1. Books and Movies Search structure

The basic idea of the Books and Movie Search was simple; the idea behind it was to find a way to use API and gather information for books and movies. This information will help users find the essential information for them and help them choose the proper film to watch or the right book. The results given in a few seconds allow people to get the information quickly and ensure that before the users read or watch the content, they get some background information. I coded my program using Flutter in Android Studio. I used my computer, the Android Studio, and some Youtube help to create the splash screen. I used an art template to connect things with the app I started. Since the template I first used as a food delivery template, I had to change several things, including changing the images of food and calories and changing the screens from delivery and service tabs to search engines. Besides that, I tried to make sure that my app looked nice, so I drew the logo myself, as shown on the first screen. Then, I tried to make some connections between my app and the other apps I had been using, and I found out that they usually have a welcome screen and allow people to enter their names. My logo is on the top of the first welcome page, showing half of the popcorn and half of the book. The logo's meaning is to allow people to gain information about books or movies.

On the second screen, there is the book search engine. To get automatic information, I tried to research using API or the online database and include this in my app. After a few tries, I found the right one and decided to connect the API with my coding. I tried to limit the search results to only the most crucial things, including the image of the book, the author, the summary, and the language that this book is in. I wanted to use this to help the users limit their reading time and see if they are interested in the book's content.

On the other hand, I used almost the same things on different search engines, including the movie search. I tried to include the language, the producer, and the overview of the whole movie. I wanted to ensure that the users would get the most basic information, allowing them to gain some background knowledge and ensure people get the information they need in a short amount of time.

I created an app that interconnects information with API to provide information for people, allowing them to get background information about movies and books quickly.

```
var _appBar = Align(
  alignment: Alignment.centerRight,
  child: Padding(
    padding: const EdgeInsets.only(top: 37.0, left: 20.0, right: 15.0),
    child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: <Widget>[
          Text(
            "Book & Movie Search",
            style: TextStyle(
                fontFamily: "Sofia",
                fontWeight: FontWeight.w800,
                fontSize: 30.0,
                letterSpacing: 1.5,
                color: Colors.white),
          ),
          Padding(
            padding: const EdgeInsets.only(top: 10.0, right: 10.0),
            child: Container(
              height: 50.0,
              width: 50.0,
              decoration: BoxDecoration(
                  image: DecorationImage(
                      image: CachedNetworkImageProvider(
                        "https://images2.imgbox.com/7d/50/GDU0vQnM_o.png",
                        errorListener: () => new Icon(Icons.error),
                      ),
                      fit: BoxFit.cover),
                  borderRadius: BorderRadius.all(Radius.circular(150.0))),
            ),
          ),
        ]),
  ),
);
```

```
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
          image: DecorationImage(
              image: AssetImage(
                  "assets/Template1/image/SplashScreenTemplate1.png"),
              fit: BoxFit.cover)),  // DecorationImage, BoxDecoration
      child: Center(
        child: Padding(
          padding: const EdgeInsets.only(bottom: 60.0),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: <Widget>[
              Image.asset(
                "assets/Template1/image/icon3.png",
                height: 45.0,
              ),  // Image.asset
              SizedBox(
                width: 10.0,
              ),  // SizedBox
              Text(
                "Movie & Book Search",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 25.0,
                    fontWeight: FontWeight.w200,
                    letterSpacing: 1,
                    fontFamily: "Poppins"),  // TextStyle
              ),  // Text
            ],  // <Widget>[]
```

```
class _bottomNavBarState extends State<bottomNavBar> {
  int currentIndex = 0;
  bool _color = true;
  Widget callPage(int current) {
    switch (current) {
      case 0:
        return new HomeScreenT1();
        break;
      case 1:
        return new BookSearch();
        break;
      case 2:
        return new MovieScreen();
        break;
      default:
        return new HomeScreenT1();
    }
  }
```

4. EXPERIMENTS/EVALUATION

In the first experiment, I tested the consistency of the result since my favorite part of the app was the images it showed; I wanted to ensure that there would be images on the results every time someone entered a name for a book or movie. To test that, I picked five films and five books. Then, I input the characters into the two different search engines to see if there will be images showing every time I search for something. I expected it to work, but it only worked for movies, not books. Then, I realized that I might have made a mistake on the app, so I went to the Android Studio and checked my codes on the book search engine. I realized I forgot to put an extra comma after the codes, resulting in a mistake. Therefore, I added another comma after the code. However, immediately after changing the codes, when I tried to reenter the thing again, the images were still not showing. I was very nervous about it and tried to ensure that nothing was wrong with the other codes. After making sure that all the other codes were correct, I thought there might be other bad things with my program. I restarted the app several times to ensure that the app worked properly, and after the third trial, images were showing again when I input the names for books.

This experiment shows that the results I am getting are consistent, and after the change in coding, the images are now displaying correctly.

The second experiment I used was to ensure that the app would not crash with many searches. I wanted to ensure that the app would work typically even after a lot of inquiries so that the app would work properly. I would not want the app to crash when the number of users increases; it will still be unknown how many people are using the app simultaneously. Therefore, I have to come up with something that works, and I need to test the effect with the app before more people use it. I want to ensure that things work correctly and that people do not need to worry about the number of users that use the app simultaneously and are worried about crashing. I wanted to ensure that the app would still work correctly even after a more significant number of searches. Therefore, I wanted to test the app and see whether the app would crash or function properly after a large number of searches. To test the app's efficiency, I decided to try twenty movies and twenty books simultaneously. To do that, I first selected the books and the movies, and I started to input the names of books and films. The number of books and movies I wanted to do was twenty, but later I thought twenty was not a more significant number, so I decided to ask more participants to join this experiment. Later, I asked my brother and my mother to enter the names of books and movies, and at the same time, after 60 searches, twenty from me, twenty from my mother, and twenty from my brother, the app still works properly. I thought that it was good to conclude that the app will be able to work correctly with an increasing number of searches. However, sixty was not a very large number, so I decided to make my grandparents and my dad test it as well. Again, after one hundred and twenty searches, the app still works properly, so I think it means that the experiment I wanted to test was on the app's effect.

I faced challenges as I tried to do the experiments; first of all, it was hard to decide which ones to test since there are multiple things the search engines will provide, and it is hard to pick a specific one for me to test. I try the images because it is always great to give some graphics in the app to make it more interesting. I wanted to create an app that others could love, and I did not want this feature to crash. So I tested it several times, surprising that the images crashed for the book search and that I needed to change the codes and ensure that it worked.

The other experiment that I did was to test whether the app would crash or not. I think it was a little challenging for the number of participants I needed to gather. I needed to ensure that the app would not crash even if multiple people were using the app at the same time. However, it was hard for my grandparents to download the app, and since we did not live together, it was hard for

me to explain things to them. It took me a long time to contact my grandparents to explain why iI needed help from them. I needed to end up making a phone call with them and explain why it was essential to have multiple people test it.

Other than that, it was nice to see that things went as planned and that it was pleasant to see the app getting done with little things left to fix.

5. RELATED WORK

One method that exists on the market is collaborative filtering, which will help the system access people's preferences on books and movies. [1] The difference between using an API and collaborative filtering is that the API does not gather personal information and recommend books and movies to the users. It is more like a machine that gives feedback when entering an input, and the output will always stay the same(with the categories or the information the app promises to provide, such as authors, summary, and producers).

Another related method on the market regarding movie search is that people use GPS to help the users connect the background information of movies and to help them locate the nearby theatres. The main difference between the movie and book search app and The Smart Movie Recommendation[2] is that my app mainly focuses on the information I will be providing. The Smart Movie Recommendation focuses more on how to help people find the movie they like and to find theatres that connect them to the theatres.

The last similar method with my app, the Movie and Book Search, is a Netflix Rest API that gathers the users' watching habits based on the last sixty hours of the content. And based on that, the API will automatically collect information that will likely be on a similar topic from the previously watched content. The difference between the Movie and Book SErach app is that my app does not collect personal data from the users and, therefore, cannot predict what people might like or not. It only shows information to provide some background information on books and movies.

6. CONCLUSION AND FUTURE WORK

The idea behind my work was simple: I wanted to find a way to minimize people's time deciding what books and movies to read or watch; for that, choosing was very hard for me. For the app Books and Movie Search, I coded with Flutter using Android Studio. I used an online API, or database, to gather information for various information for books and movies, such as authors, producers, and the summary. Once I was at the theatre and interested in the poster and the movie name, I bought the ticket and went to watch that movie. Later on, I realized the film was filled with dirty jokes and politician sacraments, and I noticed I had chosen the wrong movie. It was possible for me to google this movie and read some background knowledge of it, and at that second, something clicked in my head. What if I create an app that gathers information for both books and movies that allows people to have background knowledge of what is going on. If one wishes to watch a film and is interested in the movie name, he can use my app and enter the movie's title. Within seconds, he will be able to get a summary of the film, and instead of watching the whole film, he limits the time to a few minutes of reading time. Based on the information provided, it gives users some background idea about what the film will be about and that the app can hopefully help decide whether or not that person is interested in the content. With the help of the app, the results within the API that I included in the app will be out in a few seconds.

It is straightforward to test whether the app works or not; say I wanted to know more about the movie Flipped, I can open the app from a mobile device and select the movie tab. Then, all I need to do is to enter the movie name, in this case, Flipped, into the search engine. After a few seconds, a long list of things will appear on the screen, including the author, publication date, and a summary of this book. This app solves time-consuming deciding on what movies or books to buy that with some background knowledge of the book, it would be easy to determine whether the content is interesting or not.

Due to the fact that this app's data is gathered by an API, it could be some accuracy problem that I cannot change what information would be shown. Also, since all items that I have tested are books and movies that already exist, there might be future factors that I cannot control from the generator, and there might be a possible delay in updating new information to the app.

For the app's future development, I would like to add recommendation buttons for books and movies. The app's current version allows people to know about books and films, and I wanted to create tabs that include categories of films and books, such as trending, romance, and horror. I think that will make my app a lot neater and that If I were the user, I would love to see that there are different categories of things instead of plain search engines. If I have the chance, I would also want to add some motivational quotes, of course, either from movies or books that will update daily as people open the app.

REFERENCES

[1] KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data miningAugust 2007 Pages 550–559https://doi.org/10.1145/1281192.1281252

[2] Ko, SK. et al. (2011). A Smart Movie Recommendation System. In: Smith, M.J., Salvendy, G. (eds) Human Interface and the Management of Information. Interacting with Information. Human Interface 2011. Lecture Notes in Computer Science, vol 6771. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21793-7_63

[3] Berry, S., Fazzio, S., Zhou, Y., Scott, B. and Francisco-Revilla, L. (2010), Netflix recommendations for groups. Proc. Am. Soc. Info. Sci. Tech., 47: 1-3. https://doi.org/10.1002/meet.14504701402