Project Report

Valyrian Steelers

29 October 2017

Contents

1 Abstract

In this project, we developed a NEWS app combined with the user interface of 'Tinder app' to simplify the news-reading experience for our user. We used a newsfeed ranking algorithm to determine the order of the appearance of the NEWS. The user has freedom to like a NEWS article which helps us customise the newsfeed according to the interests of the user.

2 Introduction And Motivation

- The motivation behind this project is to simplify news reading experience on our phone, letting the app learn what the user likes or dislikes to improve the news feed.
- Our Android app displays news headlines with picture and short description in a recycler view and lets the user like or dislike using a swipe.
- The feed is customised as per the users choice based on a simplified version of Facebook's EdgeRank algorithm.
- Users can view full article in the app itself or bookmark it to view later.
- Features share button to send links through Messenger, WhatsApp and other sharing methods.
- Expected time in reading the whole article aids users in deciding to read or save for later.

3 Literature Survey

- News ranking algorithm is be based on a simplified version of the one that Facebook used to customize our news feed before 2011, known as EdgeRank algorithm
- Each news piece is assigned a score depending upon recency, source, content type as preferred by the user.

4 Problem statement

Develop an Android app to display news one card at a time. Incorporate features like bookmarks, sharing through other apps. Develop algorithm to deicide the order of new articles based on data collected from user.

5 Hardware And Software Requirements

5.1 Hardware

- Fully Functional Android Phone.

5.2 Software

- Android Studio SDK 2.3.3 with inbuilt Android phone emulator.
- Git for version control.

6 Implementation

JSON data from newsapi.org is formatted to be displayed in recycler view with each card containing a headline, a photo, a short description and expected time for reading the article.

For getting the idea of interests of our user we store a list of tokens (keywords from the headline of the News which he/she had liked earlier) along with their count and the source of the tokens (Source/Site of the News which he has liked earlier) with count.

Whenever the user likes/dislikes the app, we scan the headline of the News and tokenize it and will search for the tokens in our previous list of tokens.

If it is already there, then we increase its count(or decrease in case of unlike), else we'll add the new token in the list. The same goes for the list of the sources. In this way, we'll get the rank of interests of the user.

So the score/priority given to a news article is based on -

1. The time of the publication of the News. Higher scores is given to the feed with latest time.
2. The source of the News. If the News is from that site which our user usually like then, higher score is alloted to that NewsFeed.
3. The count of each of the tokens of the headlines which are already present in our tokens'list. The more, the merrier.
4. Score of a news article also gets updated when the user reads the article. Its score decreases when the user clicks an article and the web page loads completely. So, an article which has been read will not come on top from next time.
5. We also increase score of th newsarticles which come under the head whose checkboxes are selected by the user in the beginning or later manually.

7 Feasibility

- App is useful for those who wish to filter out irrelevant news pieces and articles.
- Users could decide whether or not to read the entire article based on estimate of time needed.

8 Features Delivered

We have delivered all features promised in the Project Proposal.

- We have used **Retrofit** to get News Api from newsapi.org in Json format and used Gson library to convert them into java objects and displayed them in the card view.
- **Glide** library is used to load image in the card view.
- We have used **Jsoup** library to convert html source code of the articles into document removing tags, css styling and javascript codes and other useless snippets. Then estimated the time to read an article from average reading speed of a person by counting number of words in the converted document.

9 Limitations

One noticeable issue is that it takes some time to parse the article's source HTML file using **jsoup** to get the word count of the source article.That's why we have we have calculated the estimated time in the background. So, 'estimated time to read' appears some time after news feed loads.

10 References

[1] https://en.wikipedia.org/wiki/EdgeRank
[2] https://techcrunch.com/2016/09/06/ultimate-guide-to-the-news-feed
[3] https://api.stackexchange.com/docs
[4] https://www.learn2crack.com/2016/02/custom-swipe-recyclerview.html
[5] https://www.developer.android.com
[6] https://www.stackoverflow.com
[7] https://www.jsonschema2pojo.org/
[8] https://www.youtube.com/watch?v=NdtE_1u0cq4
[9] https://www.safaribooksonline.com/library/view/learning-path-android/9781788292993/