Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 160 (2019) 610–615

Procedia Computer Science

www.elsevier.com/locate/procedia

The 6th International Symposium on Emerging Information, Communication and Networks
(EICN 2019)
November 4-7, 2019, Coimbra, Portugal

Personalized Micro-Service Recommendation System for Online News

Marchela Asenova[a], Christos Chrysoulas[a,*]

[a]*School of Engineering, London South Bank University, London, United Kingdom*

Abstract

In the era of artificial intelligence and high technology advance our life is dependent on them in every aspect. The dynamic environment forces us to plan our time with conscious and every minute is valuable. To help individuals and corporations see information that is only relevant to them, recommendation systems are in place. Popular platforms that such as Amazon, Ebay, Netflix, YouTube, make use of advanced recommendation systems to better serve the needed of their users. This research paper gives insight of building a microservice recommendation system for online news. Research in recommendation systems is mainly focused on improving user's experience based mainly on personalization information, such as preferences, and searching history. To determine the initial preferences of a user an initial menu of topics/themes is provided for the user to choose from. In order to reflect as precise as possible the searching interests regarding news of user, all of his interactions are thoroughly recorded and in depth analyzed, based on advanced machine learning techniques, when adjusting the news topics, the user is interested for. Based on the aforementioned approach, a personalized recommendation system for online news has been developed. Existing techniques has been researched and evaluated to aid the decision about picking the best approach for the software to be implemented. Frameworks/technologies used for the development are Java 8, Spring boot, Spring MVC, Maven and MongoDB.

Keywords: recommendation engine, collaborative filtering, recommendation phases, TF-IDF, cosine similarity

1. Introduction

Search engines obtain information on the Internet with countless results and finding the relevant information becomes a challenge. Sometimes great amount of time can be spent to locate the result expected. To reduce the time spent to pinpoint the right data, microservice recommendation systems are put in place. The use of referral systems

* Corresponding author. Tel.: +44-0-07-857473; fax: +44-0-07-857473
E-mail address: chrysouc@lsbu.ac.uk

10.1016/j.procs.2019.11.039

is beneficial not only to users, it is also beneficial to corporations that use them as a means of increasing sales. They find the application in both the e-commerce sector and in radically different sectors such as filtering news for delivery to the individual consumer. Objective of the project is to be built a software/microservice for filtering online news. The key feature of it is to filter, sort and evaluate for short period of time all the articles in the digital environment. The accurate outcome of the news exploration is not the only benefit that the system will provide, a short response time is another. A microservice is to be beneficial for the end users with time saving because the software is designed to visualise only the exact results that the user is keen on.

The rest of paper is structured as following. In Section 2 related work and approaches are discussed. The proposed system architecture is in detail presented in Section 3. Section 4 provides the needed for the implementation information. Section 5 concludes our work.

2. State of the Art - Related work

2.1. Content-based recommendation System

Content-Based Recommendation uses history available about the user behaviour. By summarizing it, it builds a user profile according to its reading history; the technique is called 'user profiling' [5]. Another similar type is 'stationary profiling' where the user's profile is provided and the affinity between the account and the new articles is calculated. The user is given the highest rated articles. This type of system hunts for the latest published articles that are analogous to the topics the user reads. The topic of the news is represented by using a vector space model such as TF-IDF [12], or the distribution of topics through language models such as PLSI and LDA [6], an analysis and assessment of similarities between new publications.

2.2. Collaborative Filtering

Collaborative filtering is another appearance of recommendation system. It works on a principle that uses user ratings for each news item to deliver a recommended feature [2], [14]. The news story is binary as every click on a news item matches a rating of 1, and if the news is not rated respectively 0. Most of their algorithms uses techniques to study user behaviour or analyse their profile. Information for each individual user is populated in order to identify his / her interests. A history of previous ratings is used to predict the behaviour of the user and its likely future interests and actions. It uses groups of users whose interests are homogenous to the interests of a user to predict new ratings. Collaborative filtering effectively evaluates user conduct when using a previous history of user behavior.

2.3. Demographic Recommender

Demographic recommender is a type of system that aims to sort users based on their personal data. The result is the recommendation of products based on demographic classes [1]. Users are divided into different groups according to demographic aspect and personal information. The purpose of this technique is to find groups of people who like a product and recommend it to another individual user who belongs to the same group of users. The user information is collected through a survey that they complete while registering or the information may be taken from the user's purchase history.

2.4. Utility-based Recommendation System

Utility-based recommendation systems recommend products based on assistive tools for each user item. The approach uses a useful feature with multiple percent attributes to determine the interests of each user and use the function to calculate the benefit of user items [3]. The problem arises is that a significant amount of interaction is required. Many of this type of systems are based on consumer product ratings. They are built on MAUT [4], by requiring considerable effort on the part of the user. Building a utility-based system with little user effort is a critical issue in its design. MAUT is a quantitative and systematic method describing the benefits of making decisions multiple independent sites based on uncertainty and preference analysis.

2.5. Hybrid-based Recommendation System

Hybrid referral systems is a modern type of recommendation system which combines content-based and collaborative filtering. It provides recommendations based on user reviews. The results of this technique are more effective [7], [8]. Both types of recommendation systems do prediction separately and then they are combined. The usage of the both content-based and collaborative recommendation is vice versa. Research has been done on comparing the different recommendation systems. As a result, the hybrid methods can provide more accurate recommendations and can be solution of data paucity and cold start problems. An instance of hybrid recommendation system is Netflix, which is comparing the watching and searching performance of the user. Afterwards comparing the user's activities and recommends movies that have common characteristics with movies that are top rated.

2.6. Knowledge-based Recommendation System

Knowledge-based recommendation engines offers product based on conclusions for user needs and interests, or in other words functional knowledge of how a product satisfies the user's needs [9], [10]. Model of the user can be any knowledge structure that supports these conclusions as preferred features for a product; and case-based reasoning system; an adapted metric (for matching) and part of an ontology. The system requires a great effort to gather knowledge, presentation and design. The described system does not depend on large amount of statistical data about top items feedback or particular users. This system helps users discover and understand an information space. There is a need to have essential knowledge about the collection of products and an informal knowledge of one's needs.

3. System Architecture

The core components of the proposed system can be found in Fig.1 are:

- User recommendations - taking care about rendering of the RSS Feeds to the user; Lenskit library [11], provides two algorithms which are applied on the existing data to accommodate a recommendation- TF-IDF and Cosine similarity [13].
- User dashboard / management page functionality– add new RSS Feeds and renders users' information; On the left-hand side paste a RSS feed URL, allocate a category for it and give ranking to determine the weight of the category and by pressing "Submit" the feed will be added, hence the user will receive notifications for news in that category; In the center of the page all news are displayed. By left click on number associated with the category, the user will be taken to a new page where the news will be displayed. On the right-hand side all feeds added are displayed. If user wants to amend the feeds needs to click on the "Edit RSS Feeds" button at the bottom of page on the left-hand side.
- User DAO- interact with the database for retrieving and manipulating the user's data; in users' documents ID for each user is allocated, his/her email – also used as identifier and a list of all registered for that user RSS Feeds pages. This data is represented by the UserRssFeed item – here the RSS Feed registered by the user is mapped with the user's rating given for that RSS Feed.

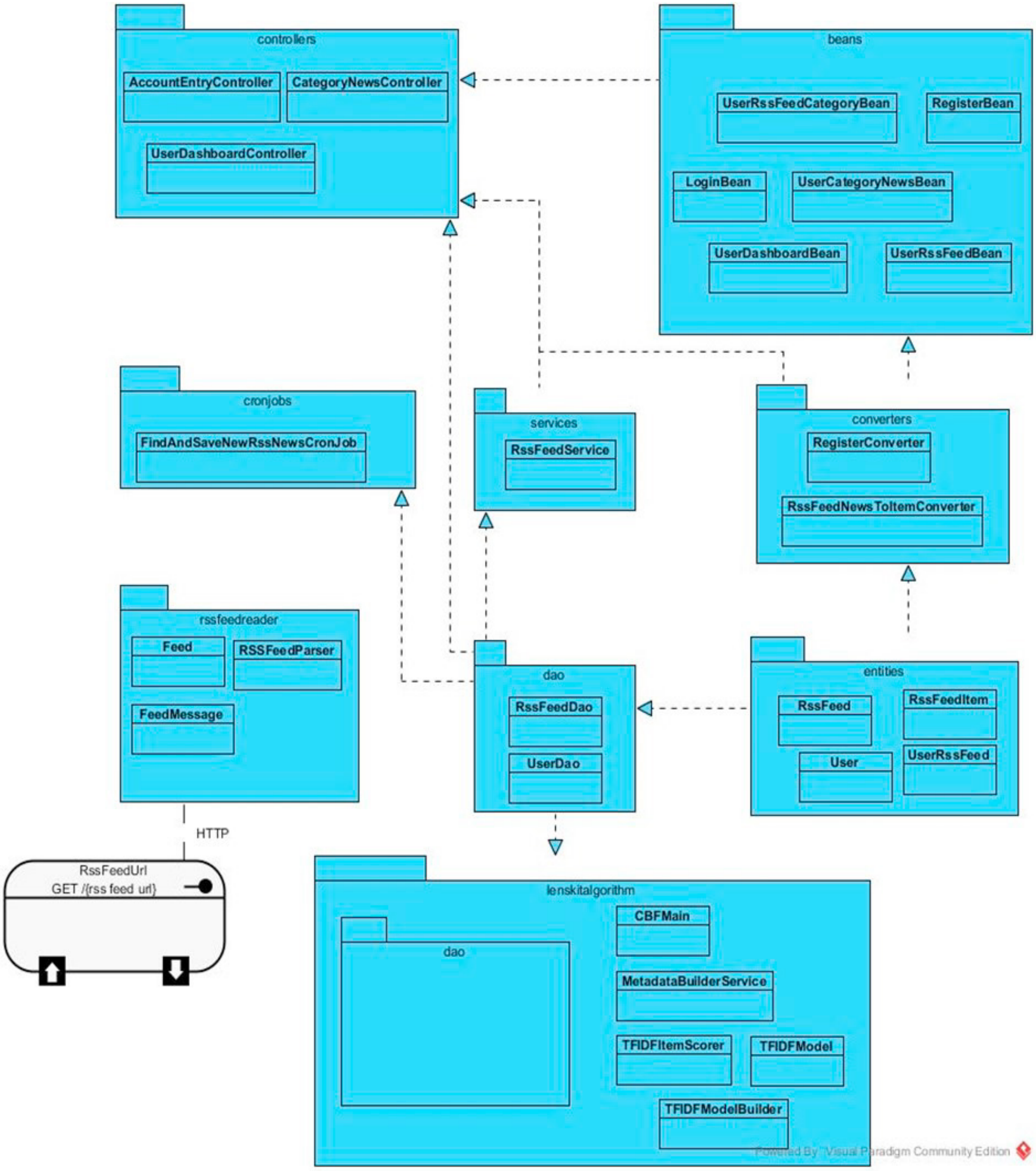

Fig. 1. System Architecture.

- RSS Feed DAO – interact with the database for retrieving and manipulating with the RSS feeds data; contains ID, a set of tags and categories – which are metadata for the RSS Feed. For example, for the RSS Feed http://feeds.bbci.co.uk/news/uk/rss.xml the metadata could be – UK, BBC, world news. Each RSS Feed contains a list from Rssfeeditem – which are actual news in the RSS Feed.
- RSS Feed extractor or the RSS Feed reader – read a new RSS Feed from an external server and persist it into the database. The RSS resource lays on an external sever. The communication being used is via HTTP protocol.
- Lenskit algorithm – the users’ RSS Feeds recommendation algorithm;

The application is being deployed on a JVM which could be installed on any operating system. The database used is MongoDB – a document-based database. The communication protocol with it is via Mongo Wire or on more abstract level – TCP Protocol. That layer is abstract for us and we use pure Java libraries (Spring Mongo DB Framework) for manipulating the data being persisted into the DB.

4. Implementation – Discussion of Results

4.1. Information Collection Phase

The first phase of recommending is collecting information. When developing the logic in this system, there are two main factors for collecting information. One is RSS Feeds from which the system retrieves tags - words that are in the content of the URL link, title and description. The other factor is the users. About the user's system retrieves information about the categories with which the user has linked the RSS Feeds and the rating the user has voted. These data are tags and are in CSV files (data format in which the values are comma separated). The CSV files contain the data the system uses to create recommendations. Four files from this type are used: users.csv, rss-tags.csv, users_ratings.csv and rss_feeds.csv. Below is the contents of the files and a diagram of the files saved in Mongo database.

Rating for Rich Site Summary (RSS) Feed is the Feedback for the Recommendation system, logic via which the data is extracted/pick up for the TF-IDF algorithm:

- RSS Feed: The system will extract tags from URL and title
- Users : (a) Add categories for a RSS Feed selected by predefine system categories (links and RSS Feed to different tags - categories); and (b) Rate each RSS Feed

At the end csv files can be pulled out where will have:

- users.csv file with all users with their ids' and nicknames
- rss-tags.csv file where we have all RSS Feeds with different tags associated for each RSS Feed
- users_ratings.csv file where we will have all ratings for all RSS Feeds per user
- rss_feeds.csv where we have RSS Feeds with id and title of the RSS Feed and URL

As a summary there are RSS Feeds with tags associated and users from the other side with rating for their RSS Feeds. The data for the recommendation system is on a .csv format. The .csv files are brought out from the MongoDB.

For Learning phase, it is used TF-IDF algorithm where the two real-valued vectors are transformed for the purposes of this report as:

- Item-tag vectors: (a) RSS Feeds and their tags associated by the users (taken by predefined categories - tags from the system) and the system (taken by the URL and the title); and (b) compute the unit normalized TF-IDF vector for each RSS Feed available in the database
- Users' vectors: (a) a vector represents the user's profile; (b) each weighted profile is computed with weighted sum of the item vectors (RSS Feed vectors) where weights are the used by the user ratings for the RSS Feeds; and (c) the Feeds are sorted by their importance
- TF-IDF Algorithm Cosine similarity.

4.2. Recommendation Phase

Each item rating is scored by using cosine similarity: the score for an item is the cosine between that item's tag vector and the user's profile vector.

Then obtain the cosine similarity of our pair of vectors by taking their dotted product and dividing that by the product of their norms. That yields the cosine of the angle between the vectors. The TF-IDF vectors are set up as:

- UPV - user profile vector
- RTV - RSS Feed Tag vector

The formula (1) used is as follow:

$\cos Q = UPV \,.\, RTV \;/\; |UPV| * |RTV|$ (REF) (1)

where Q is the angle between the vectors. As Q ranges from Q to 90 degrees, cos Q ranges from 1 to 0. θ can only range from 0 to 90 degrees, because TF-IDF's vectors are non-negative. Having in place both phases – learning and the recommendation phase and the employment of the TF-IDF algorithm, the application successfully implements the major requirements of the project – filtering, evaluation and recommendation of personalized information (news) to the user (via usage of the RSS feeds news.

5. Conclusions

The built-in referral system is found to perform as expected. All of the pre-planned functionality has been implemented. The user can use the Linux operating system application as well as any other. The only requirement to run it is to have or install the Java runtime and framework Spring Boot. The user can register / enter, adds RSS Feeds of his / her choice or from his system recommended places a rating on RSS Feed, receives news updates, updates the system every 10 minutes, the number of news for each category is numbered, opening the title sheet of all news in a category for the user, indication of the articles has not been read yet, so he knows what news are new to him; a functionality has been made to return the user to the previous pages if he wants to go back. Future work will include more additions for better facilitating the user needs and even more feature additions to search engine integration.

References

[1] Krulwich, B. (1997) "Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data", AI Magazine, Vol. 18, Issue 2.

[2] Margaritis, K. and Vozalis, M. (2006) "Web Intelligence and Agent Systems", *An international journal 4, 117–13 IOS Press*.

[3] Feng, D. (2015) "Utility-based Recommender Systems Using Implicit Utility and Genetic Algorithm", International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC 2015).

[4] Schmitt, C., Dengler, D., Bauer, M. (2002) "The MAUT-Machine: An Adaptive Recommender System." In *Proc. of the Workshop on Adaptivitat und Benutzermodellierung in Interaktiven Softwaresystemen(ABIS)*. Hannover, Germany, 2002.

[5] Bindra, A. (2012) "*SocialLDA:Scalable Topic Modeling in Social Networks*." Master of Science University of Washington.

[6] Gerlach, M., Peixoto, T., and Altmann, E. (2018) "A network approach to topic models", *Science Advances*, vol. 4, no. 7, eaaq1360.

[7] Fang, Z., Zhang, L., and Chen, K. (2016) "*Hybrid Recommender System Based On Personal Behavior Mining*." https://arxiv.org/abs/1607.02754.

[8] Burke, R. (2002) "Hybrid Recommender Systems: Survey and Experiments", in *journal of User Modeling and User-Adapted Interaction* no. 12, 2002.

[9] Burke, R. (2000) "*Knowledge-based recommender systems", Encyclopedia of Library and Information Systems, A. Kent, ed., vol. 69, Supplement 32, Marcel Dekker, 2000*.

[10] Bouraga, S., Jureta, I., Faulkner, S., and Herssens, C. (2014) "Knowledge-Based Recommendation Systems", *International Journal of Intelligent Information Technologies*, vol. 10, no. 2, pp. 1-19, 2014.

[11] "LensKit —", *Lenskit.org*, 2019. [Online]. Available: https://lenskit.org/.

[12] Qaiser, S., and Ali, R. (2018) "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents", *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25-29, 2018.

[13] Rahutomo, F., Kitasuka, T., and Aritsugi, M. (2017) "Semantic Cosine Similarity". [Online]. Available: https://www.academia.edu/Documents/in/Cosine_Similarity.

[14] Chrysoulas, C., Fasli, M. (2017) "Building an adaptive e-learning system", 9th International Conference on Computer Supported Education, 21-23 April 2017, Porto, Portugal.