iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.36128

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Personalized Event Recommendation System

Vikram Bhavsar[1], Mohammed Anzal Shaikh[2], Vishvesh Pandey[3], Samiksha Gupta[4], Prof. Rohini Nair[5]
[1, 2, 3, 4,5]*Computer Department, K.J. Somaiya University*

Abstract: Many times, we receive a large number of notifications about various events, exhibitions, and meetups happening all around us that are irrelevant to us because we simply are not interested in them. Various people have their importance of things that they are interested in and be notified of all these events and from them searching for something that might interest them will take a lot of time and sometimes does not provide any meaningful information. In today's world, there is no such existing facility that notifies us about the various events that are tailored to our interest strictly based on our web browsing history. Thus, we aim to create a Personalized Event Recommendation System that recommends the events that are sorted according to the user based on his/her interests using their browser history.

Keywords: personalized recommendation, user history, cosine similarity, tf-idf, web extension, event recommendation.

I. INTRODUCTION

We aim to create a Personalized Event Recommendation algorithm that recommends the events that are sorted according to the user based on his/her interests. The system would intend to save time for the user and also help the event organizers to reach many users and ensure that the user does not get recommended any events that he/she is not going to attend but only the events according to his/her domain-interest. The algorithm explained in this paper can be used along with other recommendation techniques [5] to provide a more robust and hybrid recommendation system. This system will have two participants-Event Organizers and Users. The idea behind this recommendation is that every user leaves a large amount of footprint when browsing various websites online. So we propose to make use of these footprints to provide better recommendations. For example: If a user on youtube is searching for Cake Baking videos it would be noted in the history. The system would then extract the title of that video and time and prepare a user profile based on similar searches. In the future when a new event is registered and that happens to be a workshop on cake baking then this event would be recommended to the user since that's one of his/her interests. The system architecture that we have designed is a website. History is taken from the user browser and sent to the server where the data is processed and recommendations are calculated and the results are shown back. A complete system can be deployed in a college where the algorithm can be applied to various intra and inter college events.

II. LITERATURE SURVEY

The algorithm that is proposed in this paper is an extension of the paper "Mining User Interests from Web History" by Saurabh Kumar [1]. The paper in depth explains how various keywords from the user's browsing history can be extracted and these keywords would be the presumed interest of the user. In our system, a client side browser extension is responsible for extracting the history from the browser and sending it back to the server. From this history the web page title and time when it was accessed is important to us. We pre-process the history and remove stop words from it, removing numerical values and converting every uppercase to lowercase to remove any ambiguity between the words. After preprocessing ranking is done which is explained in greater detail in [1]. For simplicity we have only taken a part of the process till ranking. Semantic Clustering and extracting bigrams are ignored for now. Each of the keywords is matched with the timestamp. The timestamp is the time when that particular keyword occurred. There may be a single keyword that would have multiple timestamps and that's the point. Keywords that are repeatedly found are of importance to us. Timestamp and time details are required to count the spread of that keyword. The understanding is that if repeated words have more spread it means that is something a user is interested in. If the spread is less but the count is more then it's just noise that the user was searching for or came across a topic for a day. For example: if a user is looking to buy chairs and searches about it for a single day but many times the count of the word chairs would be a lot but then again there isn't much "spread" of time between their access. They were accessed consistently and repeatedly in a short period which does not amount to something that the user may be interested in for a long time. On the other hand, if the user searches about valorant game one day, then after two-three days and then again after a week, here not only the individual count rise but also the "spread" is more and can be implied that since user is interested in the topic he/she is regularly invested to research about that topic. Any person interested in dance would look up dance videos and workshops regularly. Spread here would be more.

After the spread is calculated these keywords are ranked in descending order because the higher the spread score, the more important the keyword is to the user.

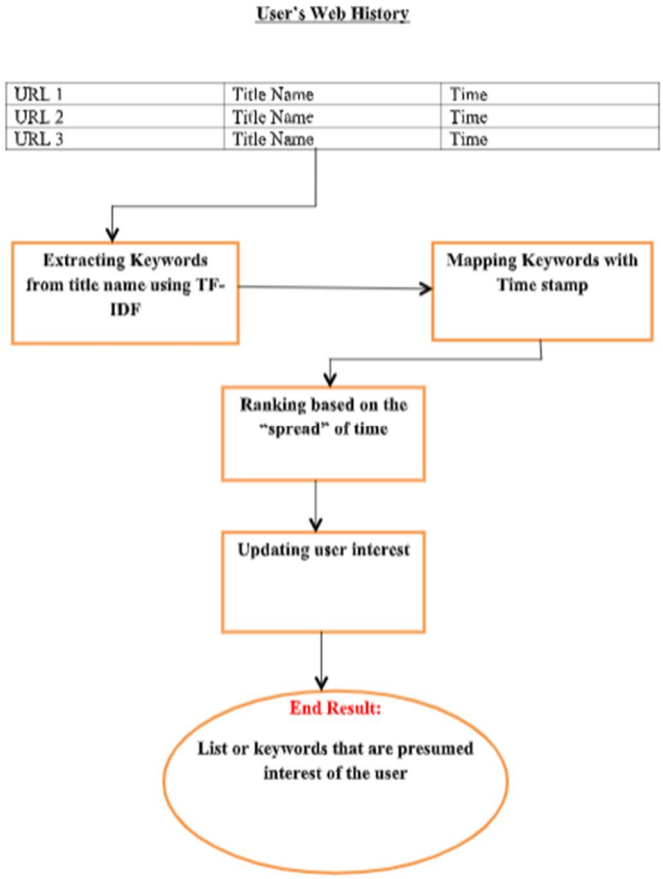


Fig. 1 Process/flow to extract keywords from user's web history

III. PROPOSED SYSTEM

A. *Extracting keywords from event's description using TF*IDF*

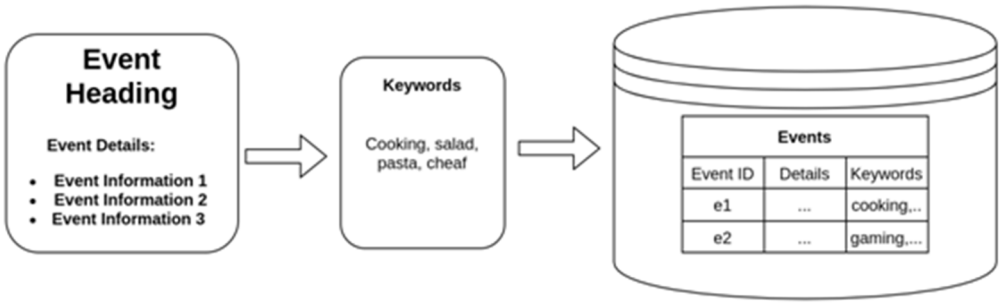


Fig 2: Keywords storing based on particular events

For providing recommendations to the user, a huge amount of Event data is needed. In an implemented system this data would be provided by Event organizers themselves. This data would include information such as description, title, date, location and other information pertaining to that event. We need to extract keywords from this event data that explains what the event is about. And hence we can apply the TF*IDF algorithm to the event title and event description together. Other details about the event can be used for other types of recommendations [5]. Here this data would need to be preprocessed similar to how the history was pre-processed. The result would be a series of keywords and their respective weight. The weight defines how important a particular keyword is. More weight means more important and less weight means less important keywords. It would be better to store keywords and their respective weight in the database so that these can be accessed easily.

B. Calculating Similarity of Events

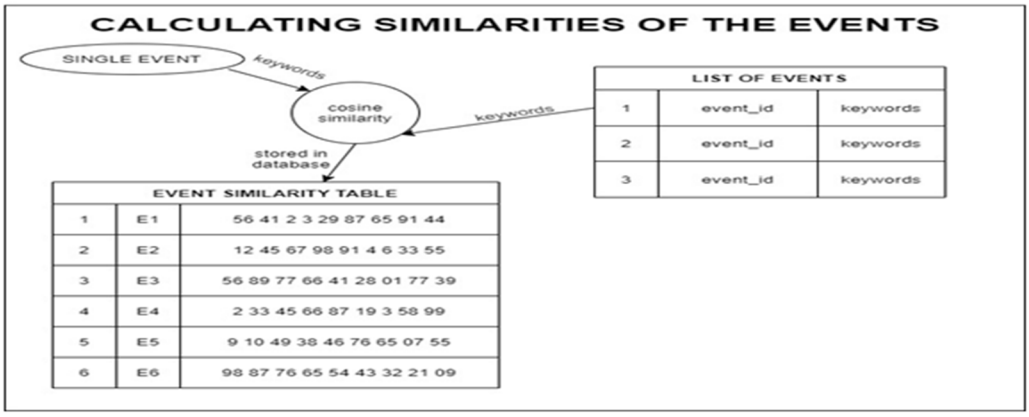


Fig 3 : Calculating Similarity of Events

Since we already have each event's keyword and their respective weight so we can calculate similarities between events in our database. We use Cosine similarity to do that. A prior calculation needs to be done before recommendations can be provided. To save time, calculated recommendations can be saved into the database as shown in fig 3. A separate table which contains event id of each event and id of other events that are similar to that event in descending order. To calculate similarity between events, keywords of both of these events can be combined together to form a matrix along with their weights. More detailed explanation on how this is calculated is provided in [4]. Cosine similarity of each event can be calculated with every other event and stored inside the database in descending order. This is certainly helpful when any interested user is viewing any event then the website can have a section below which says "browser other similar events" and top 5 or 10 events similar to that can be shown for the user to explore.

C. Extracting User's Web browsing history

We need a user's history in order to provide better recommendations based on their web browsing history. This is a complex logic and hence we need help with a browser extension because JavaScript has more power as a browser extension compared to normal web pages. We need access to the user's web browsing history which is simply not possible using normal JavaScript that we pass during serving requests.

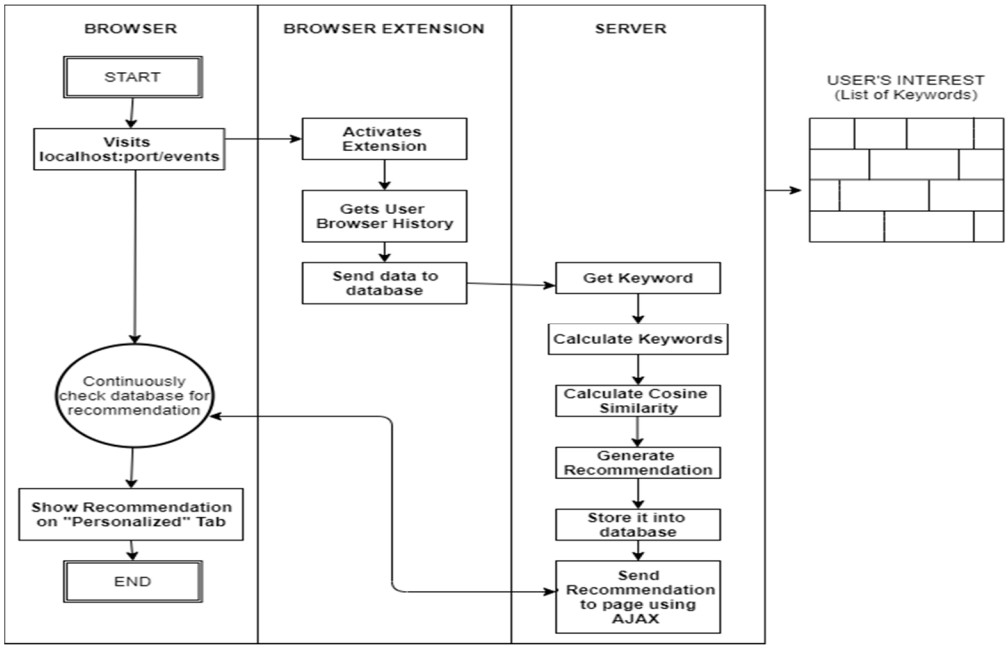


Fig 4 : Architecture of Extracting User's Web History

Whenever a client visits the link of the website. let's say localhost: port/events the browser extension that is installed is automatically activated. The task of browser extension is to gather the user's browsing history into JSON data and then send it to the server. The browser extension also extracts User ID from the page visited which is a hidden field inside in the main page. If the hidden field is missing that means the user is not logged in and hence the extension won't send any data. While at the same time the front end webpage i.e /events continuously sends requests to the server checking if recommendations are available with respect to the user's browsing history. When the browser extension has gathered all the History data it then sends it to the server. The Server calculates the recommendation and stores it temporarily inside the database. Here the user's privacy is maintained since the entire history is not stored at the server side. The history as soon as it's taken, calculations are made and only keywords are stored. For even better security, the history data can be encrypted when transferring them from the web browser to the server. When the data is ready the client side which is continuously checking the database for new records using ajax finds new records. Hence server replies with the new recommendation and the javascript on the front end render that on the client side access last access time two-three

D. Cosine Similarity Between user web History and Events.

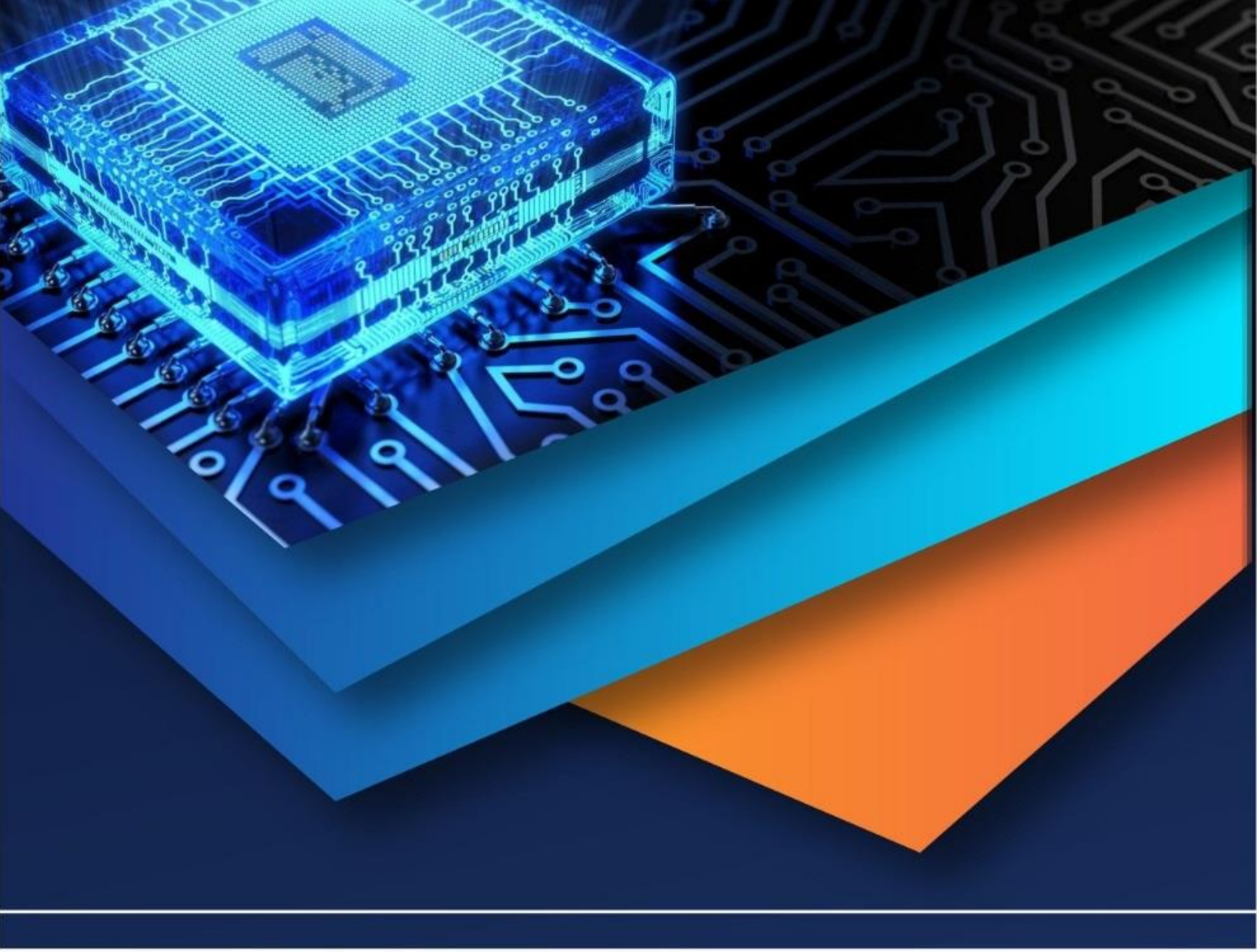
Finally at this stage we calculate the recommendation that needs to be shown to the user. We have keywords from the user's web history and respective weight of each keyword and at the same time we have keywords of each event and their respective weight. To provide recommendations based on the user's web browsing history we again calculate the cosine similarity between the user's history and each active and upcoming event that is present inside the database. This is similar to how similarity between events were calculated. Results would be a tuple of event ids and their similarity score and hence this can be sorted in descending order and provide recommendations back to the user.

IV. CONCLUSION

Personalized Event Recommendation System helps users to save time and get recommendations based on their interest. This system uses concepts such as Cosine Similarity, TF-IDF algorithm which is used to mine the user's interest from their web history to recommend the events of the user's interest. The web history is extracted with the help of a web browser extension and then cleaned and processed accordingly to mine a user's interest. Along with this, other algorithms can also be used to make the recommendations more powerful i.e Recommendation based on locality or search/ browsing patterns.

REFERENCES

[1] Saurabh Kumar, Mining User's Interest from Web History, at ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT),IEEE 2013.
[2] Cai Zhi Liu , Yan-xiu Sheng , Zhi-qiang Wei And Yong-Quan Yang, Research of Text Classification Based on Improved TF-IDF Algorithm , IEEE 2018.
[3] William Scott, "Matching two documents based on TF-IDF and Cosine Similarity". Available: https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089
[4] Cosine Similarity, Jiawei Han, Available : https://www.sciencedirect.com/topics/computer-science/cosine similarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis.
[5] Kim Falk, Practical recommender System, ,Manning Publications Co. 2019



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY