International Journal of Pure and Applied Mathematics
Volume 119 No. 12 2018, 13261-13272
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

VALIDATING EFFECTIVE RESUME BASED ON EMPLOYER'S INTEREST WITH RECOMMENDATION SYSTEM

Sivaramakrishnan N, Subramaniyaswamy V*, Arunkumar S, SoundaryaRathna P

School of Computing, SASTRA Deemed University, Thanjavur
*Corresponding Author

Abstract: In current technological world, recruitment process of corporate has evolved to the greater extent. Both the candidates and the recruiters prefer resumes to be submitted as an e-document. Validating those resumes manually is not much flexible and effective and time saving. The team requires more man power to scrutinize the resumes of the candidates. The aim of our work is to help the recruiters to find the most appropriate resume that match all their requirements. The system allows the recruiter to post his/her requirement as query, and the system will recommend the relevant resume by calculating the similarity between the query and the resume using Vector Space Model (VSM).

Keywords: recommender system, vector space model, term frequency, content based filtering, collaborative filtering.

1. Introduction

In current day, technology has a great impact on all existing as well as newly emerging fields by developing new products to solve problems and fulfill needs of the consumers by its simplicity of usage and efficiency. Also the current recruitment process of corporate world differs from the last generation [11] where the companies receive large set of resumes as e-document. This has a greater benefit over the resumes which are received as a hard copy; where searching and monitoring the appropriate candidates are considered as a very difficult as well as a tedious process. Even when the hard copies were replaced with the e-documents, recruiters find it difficult to manage correctly and rapidly the great amount of received data. Many researchers have developed systems to overcome this difficulty using recommendation system [19-29].

Recommendation system plays an important role in many fields due to increase in the vast amount of data generated in the daily life. RS is used widely in many applications including [17] movies, restaurants, online shopping websites like Amazon, music, books and even in search queries. It is also used in recommending text documents for which similarity is calculated based on the contents of the document. There are two common [10] traditional approaches in recommendation system [30-35]. They are content based filtering and collaborative filtering. The two basic elements that appear in every recommendation systems are the user and the item [36-43].

Content based filtering: Content based approach [17] recommends items based on the users past preference of an item. The similarity of the item is calculated based on the characteristics or features of the product and its textual descriptions.

Collaborative filtering: Collaborative filtering [10] is the most popular and most widespread technique in recommendation systems. This approach recommends items to the users based on the opinion of the other users whose behavior and the preferences are similar to that of the acting user.

In this paper, we propose a content based recommendation system which recommends the relevant resumes based on the employer's query using Vector Space Model (VSM)**.**

2. Literature review

In content based recommendation system, item representation has its major role. Items are nothing but the objects in the real word like books, music, movies and restaurants [2].These items cannot be processed directly by the recommendation system. Representation of these items has an impact on the result produced by the system. Poor representation of an item will equally produce the poor results. Therefore the accuracy of representation of real world objects provides an important characteristic to the recommendation system.

This representation can be either human generated or machine generated. Although the machine generated representation provides the greater performance level, we must be able to certify the accuracy of the representation generated. Machines cannot generate their own representation for the objects like books, documents, and multimedia without relying on the fields of information retrieval, data mining and pattern recognition. The representation of item can be in two approaches [2] either as a structured data or as unstructured data.

Structured data : Structured data [2] generally refers to the data that has the specific length and format .These type of data are stored in the table format using relational database and can be retrieved using some of the query languages like SQL.

Unstructured data: Unstructured data [7] refers to the data that does not have any specific length or format. It can be well explained with the help of news articles and other large text documents which follow their own format but the text contained in the documents and articles are always unstructured. To handle these huge amounts of texts, text mining comes into action which provides computational methods for [13] automated extraction of information from these unstructured text.

As technology evolves semi-structured data also comes to the picture of representing the items. Data that is neither structured nor unstructured is called semi-structured data. It does not follow any fixed schema to represent the semi-structured data. For example: XML.

In this paper, we deal with the unstructured text where resumes can be of any format. These unstructured data needs to be transformed to the structured data so that it can be directly accessed by the recommendation system. Many new technologies are evolving to analyze and process these unstructured data.

2.1. Research objectives

In recent years many researchers have focused on extracting data from unstructured text. The resume documents contain unstructured text and extracting information from those documents are still challenging, which may have different data formats, layouts and writing styles. Many advancements and researches have been made in extracting information from resumes. Resume documents have a [18] hierarchical structure of data.

Text mining provides many information extraction methods like [13] Support Vector Machines, Hidden Markov Model etc., and researchers have used various techniques like [18] cascaded hybrid model to extract information from resumes. In this model, resumes were divided into blocks and then the detailed information is extracted from each block [18].

The techniques to extract information from resumes in HTML or even in PDF format have also emerged in recent study. Document Object Model is used to represent the HTML document. Resumes in PDF format uses the [3] hierarchical model of information extraction where the documents are classified into general bocks at first, then detailed information is extracted from each useful block. If the documents were in PDF format they not only considered the content or text contained in the document but they also preferred the document's layout structure. But still some research work focuses on extracting information as a plain text and use an appropriate algorithm to develop a knowledge base for future use.

Since the content based recommendation system are at the [1] intersection of the fields of information retrieval systems and artificial intelligence in this paper, we prefer extracting information from the resume as plain text and use information retrieval method [13] like vector space model to recommend the relevant resume from the set of documents based on the employer's query or their requirements.

3. Materials and methods

We build a system using vector space model, which represent the documents as vectors and similarity measure is used to find the relevant document from the corpus. The representation of set of documents as numeric vectors is called vector space model.

3.1. Vector Space Model (VSM)

Models like VSM are preferred by the content based recommendation systems because of its simplicity. It is one of the most commonly used techniques in text mining and information retrieval system.

Let D = {d1, d2,…, d_N} be the set of N documents which represents the corpus. A vocabulary V is built from the corpus by preprocessing the text in the document. V contains the unique terms of the corpus. Here the preprocessing refers to the word tokenization, removal of stop words like a, an, or, to, etc., and stemming of words.

To calculate the importance of word in the document term weights are used which indicates the presence or importance of the word. In case of binary retrieval model, if the term is present in the document then it is represented as 1, if not 0. Term frequency can also be used which gives you the number of occurrences of the word in the document. Then the most commonly used technique is the TF-IDF which gives more effective result than other techniques.

Queries are also represented in the same way as documents. Then the similarity is calculated using various similarity measures like inner product and cosine similarity which will recommend you the relevant document based on the query.

3.2. Similarity measure

Similarity measure is used to rank the documents by calculating the similarity between the document and the query. Commonly used measures are inner product, cosine coefficient, Jaccard coefficient etc.,

3.2.1 Inner product

Multiply the document and query vector and sum the products. Longer documents are more likely to have the query terms. So inner product is used with long documents. Let the document be d and q be the query, their similarity is represented by

$$\text{Sim}(d, q) = \sum d_i \times q_i$$

Where d_i and q_i are the corresponding document and query vectors.

3.2.2 Cosine similarity

Cosine similarity is used to calculate the similarity between the two non-zero vectors and measure a cosine of angle between the corresponding vectors. It can be given by

$$Cos\,(d, q) = \ d.q/||d||\ ||q||$$

Where the numerator represents the inner product of the corresponding vectors and the denominator represents the length of the vectors. It ranges from 0 to 1.

3.3. Resume recommendation system using VSM

An item in content based recommendation system should be of [1] structured format. Usually the structured data are represented in the form of vector. A document consists of an array of words and these words are referred as the [9] bag of words which provides the greater simplicity and also used as an efficient method in text mining. VSM is the generalization of bag of words model.

Each document from the corpus is represented as a multidimensional vector. We form a vocabulary using the unique term that are extracted from the corpus. Each unique term represents one dimension of the vector space. Term can be a single word or a sequence of words [14]. Mostly the single word terms are preferred because of its high accuracy. The number of unique terms determines the dimension of the vector space. When the unique terms increase the size of the vocabulary and the dimensionality of the vector space also increase which will lead to the greater sparsity. The above problem can be resolved by preprocessing the text extracted from the document. Preprocessing includes normalizing the text, removing stop-words like auxiliary verbs and punctuations, and reducing the word to the basic or root form through stemming or lemmatization. In same way queries will also be considered as a document and represented as a vector.

In our work, the resume used is of PDF format and the java library called PDFBox is used to extract the text from the document. Tokenization of the text and the removal of stop words from the resumes are performed. Here the user query is completely based on the skills of the candidate. So we extract only those terms that match those skills. To perform this possible skills that the candidate may possess are stored in the database. We only extract the skills that are present in the database from the candidates resume. This will really help you to reduce the dimensionality of the vector space so that the vocabulary size can also be greatly reduced with the help of skill database.

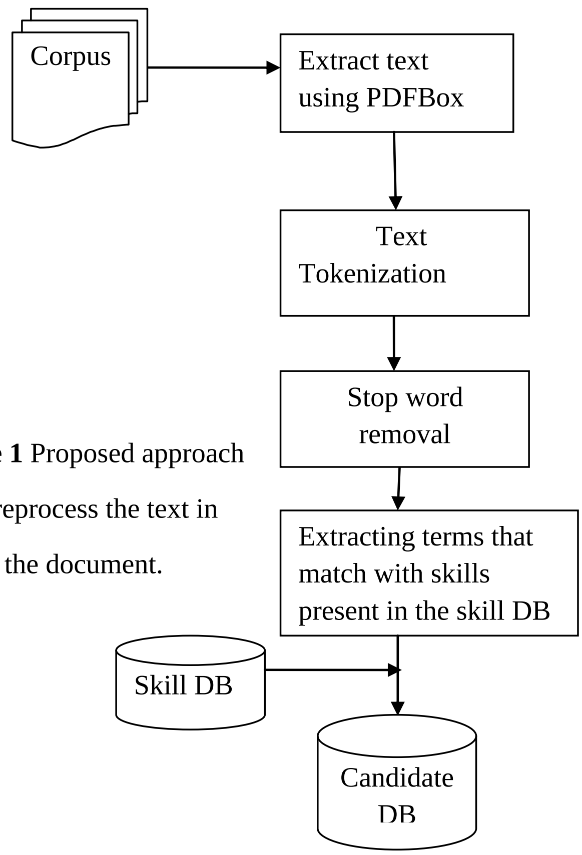

Figure 1 Proposed approach to preprocess the text in the document.

Here there may raise a problem, where the user may query a new term (considering only skills) which is not stored in the skill DB, but the document may contain the query term. This leads to the less efficient result. That is even when the document contains the query term it may not be recommended since that term has not been extracted due to its absence in the database. To avoid this we compare the query posted by the employer against the skill DB. If any new skill is posted as the query it will be stored in the skill database. By doing so we can extract all the skill terms present in the resume document. These extracted words are considered as a document and stored in the candidate DB along with the candidate’s document name and it also has the similarity attribute field which is initially zero for all the documents.

3.3.1 Architecture of the system:

Both the query and the set of skills extracted from the text are considered as a document. The documents are indexed and represented as a vector using vector space model. Here the TF weighting technique is used to count the number of occurrences of words in the document. The words that are extracted are then used to create the vocabulary which contain only the unique terms of the corpus. Vocabulary is also stored in a separate database.

Each document in the corpus is represented as the vector against the vocabulary. Query can also be represented in the same way as documents are represented. Whenever the recruiters post a query it checks if any new term other than the terms in the skill database are found. If so skill database should get updated with the new term. Here the system will search for only the newly added term and update the skills which are already stored in the candidate database. It is also necessary to update vocabulary database if the newly queried term is present in the corpus.

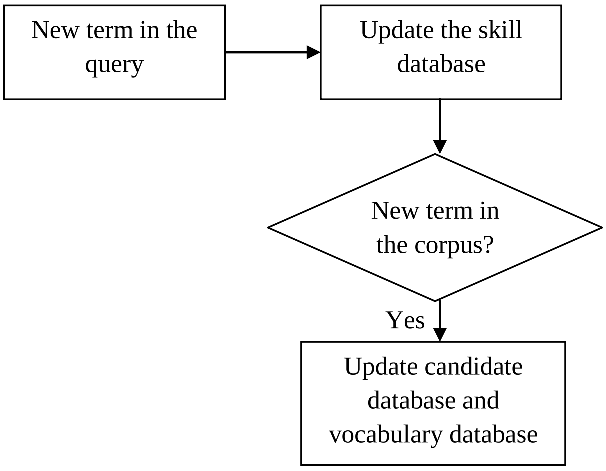

Figure 3 Handling new term in the query posted by the recruiter

Next we move on to calculating the similarity between the query vector and the document vector. Each document is compared to the query using similarity measure. Then the documents with the higher similarity value are recommended to the recruiter where the documents are ranked based on their similarity value.

Calculated similarity value is updated in the database for the retrieval of the document. We use cosine similarity to compute the similarity between the document and the query vector.

4. Experiment results and discussion

We use three database tables to store the data. They are skill DB which has the candidate's possible skill set as shown in table 1. Candidate DB which stores the name, skills of each candidate and the similarity value calculated between the document vector and the query vector which will be initially zero as shown in table 2. Vocabulary DB has the unique terms of the extracted skill in the candidate database as shown in table 3.

1. Here the corpus contains 5 documents. After the documents are preprocessed the terms that match with the skill DB are extracted and stored in the candidate database (refer table 2).

2. Then the vocabulary is constructed from the skills stored in the candidate database. Then the sorted vocabulary is stored in vocabulary database (see table 3).

Table 1 Skill DB

Skill Name
Android,CSS,Firebase,Html,Java,Spring,Python ,database,MongoDb,MySQL,nodejs,javascript, PL/SQL,SQL,J2EE,RestServices,Maven,AngularJs,Kafka,ElasticSearch,JSP,Servlet,Soap,MSSQL,Hibernate, Bootstrap,Datastructure,dreamweaver,Jquery,Oracle,eclipse,jira,Visualization,SDK,api,Struts,.net,json, apache, frameworks,XML,git,jdbc,jsp,AutomationTesting,Manual Testing,TestNG,Ajax,AWS,junit ,SDLC,Testing,Angular,Bugzilla,PHP,UNIX,LINUX,SHELL,struts,webservices,documentation , HADOOP, Integration, Agile,Interface design, Django,Machine learning,Bitcoin,MSOffice ,Spark,Hive,ASP.NET,C++,C#,Selenium

Table 2 Candidate DB

Name	Skills	Similarity
Aakash-R	Android, spring, sql, android, css, firebase, html, java, spring, python, java, android, css, html, python, javascript, firebase, java, android, firebase, android, android,	0
Abhay-Bhusari	Android, api, integration, android, json, documentation, android, android, android, android, android, android, Android, android, java, android, java, java, html, json, java, xml, android, c++, java, linux ,	0
Deepak-Kumar	Agile, soap, api, integration, java, javascript, html, css, agile, java, database, oracle, integration, api, integration, java, java, xml, javascript, integration, soap, integration,	0
Kishan-Singh	Android, android, java, html, android, c++, html, php, sql, c++, java, android,	0
Kunal-Chawda	Html, java, eclipse, database, html, java, eclipse, pl/sql, pl/sql, c++, html, java, pl/sql, android, c++, java, html, sql	0

Table 3 Vocabulary DB

Vocabulary
Agile, android, c++, css, database, documentation, eclipse, firebase, html, integration, java, javascript, json, linux, oracle, php, pl/sql, python, soap, spring, sql, xml

3. Then the skills of each candidate in the candidate database are represented as vector against the vocabulary database. Term frequency is used for the representation of vector. The vector representation of each candidate's skill is shown below.

Document 1: Aakash-R
Vector representation:
(0.0)(7.0)(0.0)(0.0)(2.0)(0.0)(0.0)(0.0)(3.0)
(2.0)(0.0)(3.0)(1.0)(0.0)(0.0)(0.0)(0.0)(0.0)
(2.0)(0.0)(2.0)(1.0)(0.0)

Document 2: Abhay-Bhusari
Vector representation:
(0.0)(10.0)(1.0)(1.0)(0.0)(0.0)(1.0)
(0.0)(0.0)(1.0)(1.0)(5.0)(0.0)(2.0)
(1.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)
(0.0)(1.0)

Document 3: Deepak-Kumar
Vector representation:
(0.0)(0.0)(3.0)(0.0)(1.0)(1.0)(0.0)(0.0)(0.0)
(1.0)(5.0)(4.0)(2.0)(0.0)(0.0)(1.0)(0.0)(0.0)
(0.0)(2.0)(0.0)(0.0)(1.0)

Document 4: Kishan-Singh
Vector representation:
(0.0)(4.0)(0.0)(2.0)(0.0)(0.0)(0.0)(0.0)(0.0)
(2.0)(0.0)(2.0)(0.0)(0.0)(0.0)(0.0)(1.0)(0.0)
(0.0)(0.0)(0.0)(1.0)(0.0)

Document 5: Kunal-Chawda
Vector representation:
(0.0)(1.0)(0.0)(2.0)(0.0)(1.0)(0.0)(2.0)(0.0)
(4.0)(0.0)(4.0)(0.0)(0.0)(0.0)(0.0)(0.0)(3.0)
(0.0)(0.0)(0.0)(1.0)(0.0)

4. Query is also considered as the document. It is also represented as the vector against the vocabulary.

Query q: java, html, javascript, css, selenium

Vector representation: (0.0)(0.0)(0.0)(0.0)(1.0)(0.0)(0.0)(0.0)(0.0)(1.0)(0.0)(1.0)(1.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)(0.0)

5. Similarity is calculated between each document vector and query vector using cosine similarity.

Cos (d1, q) = 0.4338609156373123

Cos (d2, q) = 0.25724787771376323

Cos (d3, q) = 0.5039526306789696

Cos (d4, q) = 0.3651483716701107

Cos (d5, q) = 0.

Then the documents are ranked in the descending order of their similarity values. You can retrieve the top k-most documents for any values of k.

The result for the query q is shown in table5.

Documents
Kunal-Chawda.pdf
Deepak-Kumar.pdf
Aakash-R.pdf
Kishan-Singh.pdf
Abhay-Bhusari.pdf

Table 4 Result

The documents with the higher similarity value are displayed first. And when the new term is encountered in the query we should handle the new term as in fig 2 and repeat the steps as mentioned in the above section.

5. Conclusions and future work

In this section we summarize the work done, and present the future work within the domain of resume recommendation system.

In this paper, we proposed architecture to recommend the resumes based on the employers interest. We stored the terms to be extracted from the document in the database. Only the skill attribute of the candidate is considered and extracted from the resume.

The extracted terms are also stored in the database and then the vocabulary is built from the unique terms of the corpus. Then the documents are represented as vectors using vector space model (TF method). Similarity between the documents is calculated using cosine similarity. Finally, the document with the higher similarity value is recommended to the recruiter.

The current trend of recommendation system is focusing on new methods and multidimensional objects. In this paper, we have presented how resume can be recommended using content based approach with vector space model.

We have focused only on the skills attribute of the candidate. In later works it can be extended by considering the other attributes like education, project work etc, which may raise the dimensionality of the vector space.

References

[1] Badr HSSINA1, Abdelkrim MERBOUHA2, and Belaid

BOUIKHALENE1, A Hybrid Approach of Semantic Similarity Calculation for a Content-based Recommendation of Text Documents on an E-learning Platform,(2016).

[2] Harry Zisopoulos, Content-Based Recommendation Systems (2014).

[3] Jiaze Chen, Liangcai Gao, Zhi Tang; Institute of Computer Science and Technology of Peking University; Beijing, P.R. China, Information Extraction from Resume Documents in PDF Format, (2016).

[4] M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS, Text Classification Using Machine Learning Techniques. WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp. 966-974.

[5] Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, (2003, December).

[6] Akram Roshdi1 and Akram Roohparvar2, Review: Information Retrieval Techniques and Applications. International Journal of Computer Networks and Communications Security VOL. 3, NO. 9, SEPTEMBER 2015, 373–377.

[7] Michael J. Pazzani1and Daniel Billsus2 Content-based Recommendation Systems.

[8] Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34 (1), 2002, pp. 1-47.

[9] Jelena Jovanovic, Introduction to text mining.

[10] Rebecca A. Okaka, Waweru Mwangi, George Okeyo. A Hybrid Approach for Personalized Recommender System Using Weighted TFIDF on RSS Contents. International Journal of Computer Applications Technology and Research Volume 5–Issue 12, 764-774, 2016, ISSN:-2319–8656.

[11] Khan, Tabrez Juneja, Afzal Ayub Zubeda Momin, Adnan Ayyas Shaheen Gunduka, Rakesh Narsayya Godavari Sayed, ZainulAbideen Mohd Sadiq Naseem, Resume Ranking using NLP and Machine Learning.

[12] Marko B., Yoav S. Fab: Content-based, collaborative recommendation. Communications of the Association for Computing Machinery, 1997, 40(3):66–72.

[13] Mehdi Allahyari, Seyedamin Pouriyeh, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, Krys Kochut. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques.

[14] Gerard Salton and Chistopher Buckley, Term weighting approaches in automatic text retrieval. Information Processing & Management Vol. 24, No. 5, pp. 513-523, 1988.

[15] Juan Ramos, Using TF-IDF to Determine Word Relevance in Document Queries.

[16] Jaime Arguello INLS 509: Information Retrieval Vector space model, (2013).

[17] Tulasi K. Paradarami, Nathaniel D. Bastian, Jennifer L. Wightman, A hybrid recommender system using artificial neural networks, Expert Systems With Applications 83 (2017) 300–313

[18] Kun Yu, Gang Guan, Ming Zhou, Resume Information Extraction with Cascaded Hybrid Model

[19] Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Gao, X. Z., & Indragandhi, V. (2017). A hybrid quantum-induced swarm intelligence clustering for the urban trip recommendation in smart city. Future Generation Computer Systems, 83, 653-673.

[20] Subramaniyaswamy, V., & Logesh, R. (2017). Adaptive KNN based Recommender System through Mining of User Preferences. Wireless Personal Communications, 97(2), 2229-2247.

[21] Logesh, R., & Subramaniyaswamy, V. (2017). A Reliable Point of Interest Recommendation based on Trust Relevancy between Users. Wireless Personal Communications, 97(2), 2751-2780.

[22] Logesh, R., & Subramaniyaswamy, V. (2017). Learning Recency and Inferring Associations in Location Based Social Network for Emotion Induced Point-of-Interest Recommendation. Journal of Information Science & Engineering, 33(6), 1629–1647.

[23] Subramaniyaswamy, V., Logesh, R., Abejith, M., Umasankar, S., & Umamakeswari, A. (2017). Sentiment Analysis of Tweets for Estimating Criticality and Security of Events. Journal of Organizational and End User Computing (JOEUC), 29(4), 51-71.

[24] Indragandhi, V., Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Siarry, P., & Uden, L. (2018). Multi-objective optimization and energy management in renewable based AC/DC microgrid. Computers & Electrical Engineering.

[25] Subramaniyaswamy, V., Manogaran, G., Logesh, R., Vijayakumar, V., Chilamkurti, N., Malathi, D., & Senthilselvan, N. (2018). An ontology-driven personalized food recommendation in IoT-based healthcare system. The Journal of Supercomputing, 1-33.

[26] Arunkumar, S., Subramaniyaswamy, V., & Logesh, R. (2018). Hybrid Transform based Adaptive Steganography Scheme using Support Vector Machine for Cloud Storage. Cluster Computing.

[27] Indragandhi, V., Subramaniyaswamy, V., & Logesh, R. (2017). Resources, configurations, and soft computing techniques for power management and control of PV/wind hybrid system. Renewable and Sustainable Energy Reviews, 69, 129-143.

[28] Ravi, L., & Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. Computational intelligence and neuroscience, 2016, Article ID: 1291358.

[29] Logesh, R., Subramaniyaswamy, V., Malathi, D., Senthilselvan, N., Sasikumar, A., & Saravanan, P. (2017). Dynamic particle swarm optimization for personalized recommender system based on electroencephalography feedback. Biomedical Research, 28(13), 5646-5650.

[30] Arunkumar, S., Subramaniyaswamy, V., Karthikeyan, B., Saravanan, P., & Logesh, R. (2018). Meta-data based secret image sharing application for different sized biomedical images. Biomedical Research,29.

[31] Vairavasundaram, S., Varadharajan, V., Vairavasundaram, I., & Ravi, L. (2015). Data mining-based tag recommendation system: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(3), 87-112.

[32] Logesh, R., Subramaniyaswamy, V., & Vijayakumar, V. (2018). A personalised travel recommender system utilising social network profile and accurate GPS data. Electronic Government, an International Journal, 14(1), 90-113.

[33] Vijayakumar, V., Subramaniyaswamy, V., Logesh, R., & Sivapathi, A. (2018). Effective Knowledge Based Recommeder System for Tailored Multiple Point of Interest Recommendation. International Journal of Web Portals.

[34] Subramaniyaswamy, V., Logesh, R., & Indragandhi, V. (2018). Intelligent sports commentary recommendation system for individual cricket players. International Journal of Advanced Intelligence Paradigms, 10(1-2), 103-117.

[35] Indragandhi, V., Subramaniyaswamy, V., & Logesh, R. (2017). Topological review and analysis of DC-DC boost converters. Journal of Engineering Science and Technology, 12 (6), 1541–1567.

[36] Saravanan, P., Arunkumar, S., Subramaniyaswamy, V., & Logesh, R. (2017). Enhanced web caching using bloom filter for local area networks. International Journal of Mechanical Engineering and Technology, 8(8), 211-217.

[37] Arunkumar, S., Subramaniyaswamy, V., Devika, R., & Logesh, R. (2017). Generating visually meaningful encrypted image using image splitting technique. International Journal of Mechanical Engineering and Technology, 8(8), 361–368.

[38] Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., & Vijayakumar, V. (2017). A personalised movie recommendation system based on collaborative filtering. International Journal of High Performance Computing and Networking, 10(1-2), 54-63.

[39] Senthilselvan, N., Udaya Sree, N., Medini, T., Subhakari Mounika, G., Subramaniyaswamy, V., Sivaramakrishnan, N., & Logesh, R. (2017). Keyword-aware recommender system based on user demographic attributes. International Journal of Mechanical Engineering and Technology, 8(8), 1466-1476.

[40] Subramaniyaswamy, V., Logesh, R., Vijayakumar, V., & Indragandhi, V. (2015). Automated Message Filtering System in Online Social Network. Procedia Computer Science, 50, 466-475.

[41] Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., & Indragandhi, V. (2015). Unstructured data analysis on big data using map reduce. Procedia Computer Science, 50, 456-465.

[42] Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., & Indragandhi, V. (2015). Intelligent travel recommendation system by mining attributes from community contributed photos. Procedia Computer Science, 50, 447-455.

[43] Vairavasundaram, S., & Logesh, R. (2017). Applying Semantic Relations for Automatic Topic Ontology Construction. Developments and Trends in Intelligent Technologies and Smart Systems, 48.