

Web Page Recommendation Using Distributional Recurrent Neural Network

Chaithra^{1,*}, G. M. Lingaraju² and S. Jagannatha³

¹Department of Computer Science & Engineering, Sapthagiri College of Engineering, Bangalore, India

²Department of Information Science and Engineering, Sahyadri College of Engineering & Management, Mangaluru, India

³Department of Computer Applications, MSRIT, Bangalore, India

*Corresponding Author: Chaithra. Email: chaithra81@gmail.com

Received: 17 February 2022; Accepted: 25 April 2022

Abstract: In the data retrieval process of the Data recommendation system, the matching prediction and similarity identification take place a major role in the ontology. In that, there are several methods to improve the retrieving process with improved accuracy and to reduce the searching time. Since, in the data recommendation system, this type of data searching becomes complex to search for the best matching for given query data and fails in the accuracy of the query recommendation process. To improve the performance of data validation, this paper proposed a novel model of data similarity estimation and clustering method to retrieve the relevant data with the best matching in the big data processing. In this paper advanced model of the Logarithmic Directionality Texture Pattern (LDTP) method with a Metaheuristic Pattern Searching (MPS) system was used to estimate the similarity between the query data in the entire database. The overall work was implemented for the application of the data recommendation process. These are all indexed and grouped as a cluster to form a paged format of database structure which can reduce the computation time while at the searching period. Also, with the help of a neural network, the relevancies of feature attributes in the database are predicted, and the matching index was sorted to provide the recommended data for given query data. This was achieved by using the Distributional Recurrent Neural Network (DRNN). This is an enhanced model of Neural Network technology to find the relevancy based on the correlation factor of the feature set. The training process of the DRNN classifier was carried out by estimating the correlation factor of the attributes of the dataset. These are formed as clusters and paged with proper indexing based on the MPS parameter of similarity metric. The overall performance of the proposed work can be evaluated by varying the size of the training database by 60%, 70%, and 80%. The parameters that are considered for performance analysis are Precision, Recall, F1-score and the accuracy of data retrieval, the query recommendation output, and comparison with other state-of-art methods.

Keywords: Ontology; data mining in big data; logarithmic directionality texture pattern; metaheuristic pattern searching system; distributional recurrent neural network; query recommendation



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

CLOUD is an eminent technology in recent days, which provides highly scalable services to web pages. It enables the customers to rent out the spaces on their physical machine with increased profit maximization. The cloud computing environment is classified into homogeneous and heterogeneous clouds. In a homogeneous cloud, the entire service is offered by a single vendor and in a heterogeneous cloud, the service contains components integrated from various vendors. In the query recommendation process, the Data recommendation concept was majorly focused to present the sorted list of subject and course information by referring to the database. Big data in Data recommendation focused to analyze the data cluster with labeled properties that are can be characterized based on the ratings, probability of visit, and other parameters. Web service is a platform-independent factor that is mainly used to help machine-machine communication in a network [1]. Accuracy is one of the major considerations for the web pages to select their required services, which describes the non-functional characteristics of the web services. Normally, the Prediction [2] is defined as the set of characteristics of data availability, reputation, and throughput. Normally, the processes of service selection and recommendation are the major things that are used to enable service composition in recent years [3]. The traditional works developed similarity-based query recommendation systems. It does not satisfy the web page requirement by providing the most similar services [4]. Moreover, it utilized some encryption and clustering mechanisms during the service storage and retrieval. The existing encryption techniques such as K-means [5], K-Medoids [6], and Fuzzy C-Means (FCM) [7,8] are not highly efficient for query recommendation, because, it has the major issues of being highly sensitive, requires a finite number of clusters, and large searching space. The major reasons for using the encryption [9] techniques are to ensure the confidentiality, privacy, integrity, access control, and authentication of the data. Then, the existing encryption techniques such as Elliptic Curve Cryptography (ECC), blowfish, Advanced Encryption Standard (AES), and Rivest Shamir Adleman (RSA) are used in the traditional works, which has the drawbacks of increased complexity and time consumption. So, these techniques are also not highly suitable for an accurate query recommendation [10]. The major objectives of this paper are as follows:

- To preprocess the given dataset, the stop words removal and stemming processes and extract the features of data that are arranged according to attributes.
- To for the cluster of data feature set by using the Logarithmic Directionality Texture Pattern (LDTP) method for big data.
- To find the similarity indexing by using the Metaheuristic Pattern Searching (MPS) system and arrange the data in proper paging architecture.
- To arrange the data by sorting feature attributes based on their similarity and form the hierarchical structure.
- To improve the accuracy of the query matching and improve the speed of the process.
- To recommend the most similar items to the requested web pages, the correlation factor between the data attributes is to be computed with the best matching.

The rest of the sections in the paper are organized as follows: Section 2 reviews the existing frameworks and techniques that are used for query recommendation. Section 3 provides a clear description of the proposed methodology with its detailed flow representation. The experimental results of the existing and proposed mechanisms are evaluated and compared in Section 4. Finally, the paper is concluded and the enhancements that can be implemented in the future are stated in Section 5.

2 Related Work

In this section, the existing techniques and algorithms related to query recommendation are surveyed with their advantages and disadvantages.

In [11] developed a location aware personalized collaborative filtering mechanism for improving the Prediction of the recommendation system. In this mechanism, both the location of the web pages and the web services were leveraged for electing the target service or the web page. The main Prediction factors that considered in this work were availability, response time, web page dependency, and reliability. The stages involved in this system were as,

- Web page location information
- Identification of similar web pages
- Prediction based on the web pages
- Prediction based on the services
- Prediction based on web pages and services
- Recommendation

Here, the similarity computation was performed by selecting similar neighbors with the use of the weighted PCC technique. Also, the impact of sparseness was examined in this work for evaluating the accuracy of prediction. Reference [12] analyzed the shortlists for supporting the web page decision process in a recommendation system. In this paper, it was stated that the shortlists provided a better improvement in both downstream and web page satisfaction performance. Also, it improved the quality of recommendations by implementing additional feedback. Reference [13] Suggested a context-aware Prediction scheme for the web page recommendation system. Here, the mapping relationship between the geographical distance and similarity value was analyzed in the web page side. Moreover, this mechanism selected the most effective similarity function for attaining an exact similarity between the web pages. Moreover, the Matrix Factorization (MF) method was utilized as the basic model, which offered integrated context information. The disadvantage behind this research work was, it required attaining the detailed resource configuration with a time factor. Reference [14] Developed a new approach by integrating collaborative filtering with the content-based filtering technique for service recommendation. In this system, the semantic content and rating data were utilized to perform the recommendation by the use of the probabilistic generative model. The key objective of this paper was to investigate the recent state of the art in web service recommendation systems. Moreover, three main requirements such as recommendation serendipity, recommending newly deployed services, and recommendation accuracy were examined in this work for developing an efficient recommendation system. Then, a three-way aspect model was implemented to identify the similarities of the web pages based on the semantic contents of web services. Reference [15] suggested three different recommendation approaches such as collaborative filtering approach, content based approach, and hybrid approach for developing an efficient service recommendation system. The major components involved in this system were as follows:

- Functional and non-function evaluation.
- Diversity web service ranking.
- Diversity evaluation.

To consider the query recommendation system for the Data recommendation database, [16] proposed an ontology based context recommendation system and mobile data recommendation applications. In this, the context types like Profile, Social interactions, learning activities, and device specifications were considered from the database in the learning object ontology. The OWL rules are used for the filtering of context to recommend the query input. In [17], the paper presented a survey of different methodologies for the recommendation system based on the ontology of Data recommendation. In that, it states that the hybridization of algorithms and other recommendation techniques achieved a better similarity identification model based on the knowledge based recommendation system. Later in [18], the author proposed a course recommendation system based on the query classification approach. This estimates the

relevant data for the query input by using the classification of query data from the database. The ontology estimates the N-List of relevant features from the database that are matched with the query input and displayed the recommended course information. Similarly in [19], the paperwork presented a review of ontology based Data recommendation process. The Recommender systems specified in the analysis are using ontology, artificial intelligence, among other techniques to provide personalized recommendations. This helps to prepare the learning libraries and feature retrieval model to enhance the recommendation system in the Data recommendation process. In [20], the author proposed a novel learning path recommendation model. This was based on the multidimensional knowledge graph framework for the Data recommendation system. This multidimensional knowledge graph method was used to separate the overall database and organized it into several classes. This will enhance the learning capacity and reduce the time complexity of the classification model. Similarly, [21] paperwork proposed a novel recommendation system for the Data recommendation process using the moodle Data recommendation platform. This identifies the similarity of course information from the database and retrieves the relevant data. MoodleRec performs the sorting of supported standard compliant Learning Object Repositories and suggests a ranked list of Learning Objects that are similar to the query input that is operated in the two different levels of classification. In [22] for the telemedicine diagnosis 3D imaging helps the doctors to make clear judgments, 3D medical watermarking algorithm based on wavelet transform is proposed in this work. In [23] to avoid medical audio data leakage in the field of telemedicine the two-stage reversible robust audio watermarking algorithm was proposed.

From the survey, it is investigated that the existing approaches have both advantages and disadvantages, but it mainly lacks the following drawbacks:

- It failed to recognize the Prediction variations.
- The existing systems require more training sets to retrieve query data.
- This also increased the time complexity of the memory based recommendation systems.
- It offered a list of ranked services with no transparency.

To solve these problems, this paper aims to develop a new query recommendation system. In this, the query searching and relevant data identification were processed by similarity indexing and the Logarithmic Directionality Texture Pattern (LDTP) based cluster estimation. The Distributional Recurrent Neural Network (DRNN) based classification algorithm enhanced the accuracy performance of the data retrieval system than by using the traditional recommendation model. A detailed description of the proposed work is explained in Section 3.

3 Proposed Work

In this section, a detailed description of the proposed methodology is presented with its flow illustration. This paper intends to perform query recommendations efficiently. In this system, the progressive data like student education course data is given as the input, which is preprocessed by performing the stop words removal and stemming.

After getting the filtered/pre-processed data, the matrix is generated for selecting the Cluster Pattern (CP) based on the Normalized Logarithmic Directionality Texture Pattern (LDTP) method. Based on the CP, the clusters are formed by implementing the Metaheuristic Pattern Searching (MPS) system and the documents in each cluster are then processed for the training of the classification algorithm by using Distributional Recurrent Neural Network (DRNN) as shown in Fig. 1. After that, the similarities such as Kolmogorov and Transformation distance are computed to identify the similar items. Consequently, the n numbers of attributes of the documents are stored in the cloud server. When the server receives the request from the web page, retrieve the relevant data by searching for the best match for query data and recommend the

result by using the DRNN classification model. Based on the highest similarity value, the services are recommended to the requested web page. Finally, the matched data that are related to the query input is listed as the recommended result for the Data recommendation application.

The stages that are involved in this system are as follows:

- Preprocessing
- Logarithmic Directionality Texture Pattern (LDTP)based pattern generation
- Clustering using Metaheuristic Pattern Searching (MPS)
- Similarity Estimation
- Service Ranking and recommendation

3.1 Preprocessing

At first, the dataset is given as the input for preprocessing, where the stop words removal and stemming are performed to obtain the filtered data. The main intention of dataset preprocessing is to optimize the data size by selecting attributes that are related to the recommendation process. Here, the unwanted characters or letters are filtered to reallocate the special characters that are needed to process in the dynamic analysis model. The special characters are considered as the Unicode value to represent the letter size which can reduce the data memory size. This makes the prediction quality with better retrieval accuracy. The filtering of irrelevant data can be identified by estimating the uniqueness of the attribute value whether that value can be segmented by its related components.

3.2 Logarithmic Directionality Texture Pattern (LDTP) Based Pattern Generation

After preprocessing, the matrix is generated for the preprocessed data by using the LDTP. It is also known as the universal distance measure that finds the distance between each object. Moreover, it simultaneously uncovers all similarities for selecting the CP. In this stage, the set of documents and their domain list is given as the input. Here, each domain contains a set of N files, which are stored in a repository R_N and its size is denoted as M . To construct the matrix, the varying number of keywords in each document are extracted i.e., K_i and K_j . Then, the bytes of K_i and K_j is also extracted and stored in separate variables wd_x , wd_y and wd_{xy} . Consequently, the binary values are calculated for the extracted bytes of data, based on this the values of M_{xy} , N_{xy} and N_{yx} are computed as shown below:

$$M_{xy} = \begin{cases} |B_x * B_x| & \text{if } (B_x > B_y) \\ |B_y * B_y| & \text{Otherwise} \end{cases} \quad (1)$$

$$Nk_{xy} = \begin{cases} N_{xy} & \text{if } (N_{xy} > N_{yx}) \\ N_{yx} & \text{Otherwise} \end{cases} \quad (2)$$

Then, the distance $dist_{xy}$ is computed by the ratio of $\frac{1}{\log(Derv_{xy})}$ and dif_v . Consequently, the T_{dist} is updated with the sum of T_{dist} and $dist_{xy}$.

$$dist_{xy} = 1 - \frac{(2 * \left(\frac{1}{\log(Derv_{xy})} - min_v \right))}{dif_v} \quad (3)$$

$$T_{dist} \leftarrow Update (T_{dist} + dist_{xy}) \quad (4)$$

Then, the value of O_{dist} is updated with the values of T_{dist} and finally, the $M_{NID}(i, j)$ is estimated based on the updated O_{dist}/M .

Algorithm 1: Logarithmic Directionality Texture Pattern (LDTP)

Input: Set of documents, Domain list

Output: LDTP matrix (M_{NID})

Step 1: Let, N be the set of Domains, in which each domain contains a sample set of 50 files;

Step 2: Let, R_N be of the repository of files, which holds the set of N documents;

Step 3: Let, M be the Total Size of R_N ;

Step 4: To construct the M_{NID} of size (M, M)

Step 5: For $i = 1$ to M

$K_i \leftarrow \text{extractKeyWords}(R_N(i))$

For $j = 1$ to M

$K_j \leftarrow \text{extractKeyWords}(R_N(j))$

For $x = 1$ to size (K_i)

For $y = 1$ to size (K_j)

$wd_x = \text{extract bytes}(K_i(x))$

$wd_y = \text{extract bytes}(K_j(y))$

$wd_{xy} = \text{extract bytes}(K_i(x) * K_j(y))$

$B_x = \text{compute fold}(wd_x)$

$B_y = \text{compute fold}(wd_y)$

$B_{xy} = \text{compute fold}(wd_{xy})$

Compute M_{xy} by using Eq. (1)

Compute N_{xy} , N_{yx} by using Eq. (2)

$N_{xy} = \text{dif}(B_{xy}, B_x)$

$N_{yx} = \text{dif}(B_{xy}, B_y)$

If ($N_{xy} > N_{yx}$)

$NK_{xy} = N_{xy}$

Else

$Nk_{xy} = N_{yx}$

End if;

$Derv_{xy} = M_{xy} / NK_{xy}$

The distance $dist_{xy}$ and T_{dist} are computed by using Eqs. (3) and (4);

End for x

$O_{dist} \leftarrow \text{Update}(O_{dist} + T_{dist})$

End for y

$M_{NID}(i, j) \leftarrow \text{Update}(O_{dist}/M)$;

End for i

End for j

3.3 Clustering Using Metaheuristic Pattern Searching (MPS)

After selecting the CP using LDTP, the number of clusters is formed by implementing the MPS technique. When compared to traditional clustering such as k-means and fuzzy c-means, it is an efficient clustering technique widely used in the field of computer science. Because it provides a high quality of clusters by iteratively exchanging the messages between all pairs of data. The major reasons for using this algorithm are reduced clustering error, determinism, increased efficiency, and simple computation. Also, it does not require satisfying the triangle inequality, because it supports the similarities. Moreover, the major characteristics of this technique are availability and responsibility. This algorithm computes a set of exemplars for representing the dataset, where the pair-wise similarity is estimated between each pair of data. Here, the sum of distances or similarities for all the data points with respect to their equivalent exemplars is maximal. In this algorithm, the availability matrix S_i and the responsibility matrix S_j are constructed based on the distance matrix M_{NID} obtained from the previous stage. Then, these matrices are updated by checking the rows and columns in M_{NID} is greater than the value of A_{ij} .

$$A_{ij} = \begin{cases} 0.5 & \text{if } (M_{NID}(i, j) \leq 0.5) \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

Consequently, the exponential matrix is constructed by checking the value of the sum of A_{ij} and R_{ij} is greater than 0.

$$R_{ij} = \begin{cases} M_{NID}(i, j) - A_{ij} & \text{If } (A_{ij} \leq M_{NID}(i, j)) \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

Then, the average for the $Max(Exp_m_i)$ and $R_{i(Idx_x)}$ is computed and updated with the avg_{list} .

$$Exp_{m_{ij}} = \begin{cases} 1 & \text{if } (A_{ij} + R_{ij}) > 0 \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

$$avg = \sum_{x=1}^{size(Idx_{ls})} R_{i(Idx_x)} \quad (8)$$

where i and j are the size of the matrix (S_i, S_j), $Idx_{ls} \leftarrow \text{Max}(Exp_{m_i})$ where i is the size of Matrix and S_i is the Index list with maximum elements of each column in the matrix. After that, the distance index list is estimated by finding the difference between the avg^2 and avg_x^2 . Then, the maximum index of the dis_{ls} is calculated, based on this value, the CP is selected for each cluster.

$$avg_R = \frac{\sum_{j=1}^{S_j} R_{ij}}{S_j} \quad (9)$$

$$dis_{ls} = \sqrt{avg^2 - (avg_x^2)} \quad (10)$$

where x is the size of the matrix S_j and $C_{id} = \text{Max}(\text{Index}(dis_{ls}))$.

Algorithm 2: MPS based Clustering

Input: Distance Matrix [LDTP matrix (M_{NID})]

Output: clusters, A_{ij} , R_{ij}

Step 1: Construct Availability Matrix and responsibility matrix

(Continued)

Algorithm 2 (continued)

Let S_i and S_j be the size of the matrix (M_{NID}) and Set $K = 2$;

For $i = 1$ to S_i

For $j = 1$ to S_j

A_{ij} is computed by using Eq. (5)

End for j ;

End for i ;

Step 2: Construct and update the responsibility matrix and Availability matrix;

For $X_k = 1$ to k

For $i = 1$ to S_i

For $j = 1$ to S_j

R_{ij} is computed by using Eq. (6);

End for j

End for i

For $i = 1$ to S_i

For $j = 1$ to S_j

Let $temp_R_{ij} = 0$;

For $m = 1$ to S_i

$temp_R_{ij} = temp_R_{ij} + R_{im}$

End for m ;

$$temp_R_{ij} = \begin{cases} temp_R_{ij} + R_{ij} & \text{if } (temp_{R_{ij}} \leq 0) \\ R_{ij} & \text{Otherwise} \end{cases}$$

$$A_{ij} = \begin{cases} 0 & \text{If } (i \neq j) \\ 0 & \text{If } (temp_R_{ij} < 0) \\ temp_R_{ij} & \text{Otherwise} \end{cases}$$

Else

$$A_{ij} = \begin{cases} temp_R_{ij} & \text{If } (temp_R_{ij} > 0) \\ 0 & \text{Otherwise} \end{cases}$$

End if

End for S_j

End for S_i

End for X_k

Step 3: Compute Exponential Matrix by using Eq. (7);

Update $avg_{list} \leftarrow avg$;

For $y = 1$ to S_j

Compute avg_R by using Eq. (9);

Compute the distance list by using Eq. (10);

Update $C_{id} \rightarrow C_{head}$

End for y ;

3.4 Similarity Computation

In this stage, the similarity is computed between the documents by computing the Kolmogorov and transformation distance based similarity measures. The Kolmogorov is a widely used similarity mechanism that encodes a finite set of objects into strings denoted as $\{0, 1\}$. For instance, it estimates the similarity between two representations (i.e., A and B), where it takes A as input and B as output. Then, the quantity of this similarity is denoted as the $K(B|A)$, which is semi-computable. Then, the transformation distance based similarity measure is a kind of asymmetric technique and it does not have admissible distance. In this technique, the web page query is given as the input and the estimated similarity is results as the output. Here, the size of cluster head documents and the keywords in each cluster head document is computed then the size of the document is converted into bytes. Also, the binary folds are computed for the bytes of data, from that the minimum and maximum folds are estimated by using the following equation:

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} (B_{fx} * B_{fy}) = \text{Max} (B_{xy}) \\ (B_{fy} * B_{fx}) = \text{Min} (B_{xy}) \end{array} \right. \quad \text{if } (B_{fx} > B_{fy}) \\ \left\{ \begin{array}{l} (B_{fy} * B_{fx}) = \text{Max} (B_{xy}) \\ (B_{fx} * B_{fy}) = \text{Min} (B_{xy}) \end{array} \right. \quad \text{Otherwise} \end{array} \right. \quad (11)$$

$$p1 = \left(\frac{(B_{fy} - \text{Min} (B_{xy}))}{\text{Max} (B_{xy})} \right) \quad (12)$$

$$p2 = \frac{((\text{Size} (S1) - \text{Size} (S2))}{\text{size} (S12)} \quad (13)$$

Based on these values, the transmission distance similarity is estimated by the product of $p1 * p2$. Then, the KC is computed by generating the mask value for the binary folds of the data.

$$\text{Sim}_{Kc} = \left(\frac{(K_{xy} \& \& \text{Mask}) - \text{Min} (B_{xy})}{\text{Max} (B_{xy}) - \text{Min} (B_{xy})} \right) \quad (14)$$

At last, the similarity values of both TD and KC are summed for estimating the total similarity. It is used to identify the most similar items related to the web page query.

Algorithm 3: Similarity Computation

Input: Web page Query

Output: Estimated similarity

Step 1: Let U_Q be the Web page Query

Step 2: Let K_U be the Keywords in the Web page Query

Step 3: The Server ID K_U with the cluster Keys.

Step 4: Let c_{Head} be the cluster head documents and kc_H be the keywords in the cluster head documents

For $M = 1$ to Size of (K_U)

For $N = 1$ to Size of (kc_H)

$S1 = K_U(M)$, $S2 = kc_H(N)$ and $S12 = s1 * s2$

$B1, B2$ and $B12$ bytes for of $S1, S2$ and $S12$

Bf_x, Bf_y and Bf_{xy} be the binary folds of $B1, B2$, and $B12$;

(Continued)

Algorithm 3 (continued)

Compute Max (B_{xy}) and Min (B_{xy}) by using Eq. (11);
 Compute $p1$ and $p2$ by using Eqs. (12) and (13);
 $Sim_{TCD} = p1 * p2$;
 $K_{xy} = Bf_{xy} - Min (B_{xy})$
 Set Mask = 0xFF;
 Compute Sim_{Kc} by using Eq. (14)
 $Sim_{Tot} = (Sim_{TCD} + Sim_{Kc})$
 End for N
 $C_{Ch} = Min(Index (Sim_{Tot}/N))$;
 End for M

3.5 Query Recommendation

Finally, the ranking is provided for the items based on their similarity value, where the high similar items are recommended to the requested web pages. Here, the Distributional Recurrent Neural Network (DRNN) is used to rank the items based on their similarity. In this technique, the web page query and selected CP are given as the input and the matched document for the query is provided as the output. At first, the server retrieves the keywords in the web page query K_U , then the keywords in the number of documents K_{Dn} are extracted. After that, the similarity between the web page query and the keywords in the document is computed, based on this the list of similarity score Sim_{Ku} is estimated for the number of documents in the matched cluster. Finally, the matched document E_{rank} is listed and displayed as the recommended information about the query data for the classification process.

Algorithm 4: DRNN

Input: Web page query, Chosen cluster head C_{Ch}

Output: Matched document for the Query

Step 1: Let U_Q be the Web page Query

Step 2: Let K_U be the Keywords in the Web page Query

Step 3: The Server index K_U with the cluster Keys.

Step 4: Let N be the set of documents in the cluster;

Step 5: For $I = 1$ to N (No of documents)

$K_{Dn} =$ Keywords in the Document

$Sim_{Ku} \leftarrow$ Similarity (K_{Dn}, K_U) and update

End for N

Step 6: Sim_{Ku} is the list of similarity scores for the documents in the matched cluster.

$$E_{rank} = \frac{Min(Sim_{Ku})}{\sum_{m=1}^N Sim_{Ku}}$$

The matched document is denoted as $N (E_{rank})$

Step 7: Decrypt the matched document based on the above mentioned non-abelian algorithm;

4 Result Analysis

In this section, the experimental results of existing and proposed techniques are evaluated by using various performance measures. The overall implementation of the proposed work was processed in the tool python (Version 3.7). It includes precision, recall, f-measure, and other classification rates. Moreover, the proposed service recommendation mechanism is compared with the existing similarity and classification technique for proving the effectiveness of the proposed system.

Precision, recall, and f-measure are the most used measures for evaluating the performance of the query recommendation methods. In which, precision is defined as the function of relevancy, which estimates the ratio of the relevant and retrieved services. Also, it is the positive predictive value that provides the results relevant to an accurate service recommendation. The recall also provides the most relevant results during the service recommendation. Then, the f-measure integrates the value of both precision and recall, which reduces the impact of outliers. The precision, recall, and f-measure values are calculated as follows:

$$Precision = \frac{Relevant \cap Retrived}{Retrieved} \quad (15)$$

$$Recall = \frac{Relevant \cap Retrived}{Relevant} \quad (16)$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (17)$$

Figs. 2 to 4 shows the precision, recall, and f-measure values of the proposed service recommendation system with respect to varying α values. Fig. 3 shows recall values of the proposed work. Here, α represents the coefficient that states the more specific information of the functionalities, which ranges from 0.6 to 0.9. In this evaluation, various fields such as business, entertainment, politic, sports, and technology are considered. Also, the comparison between the existing [24]. Tab. 1 shows the number of classes for different categories and the number of training documents and testing are listed for the dataset followed in the existing paper [25].

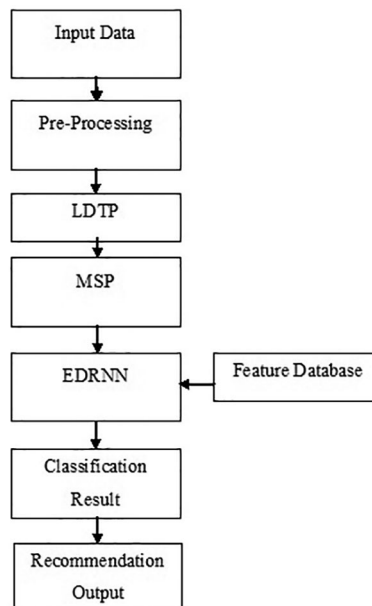


Figure 1: Overall block diagram of proposed work

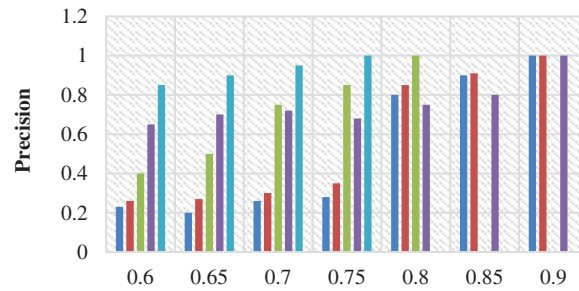


Figure 2: Precision

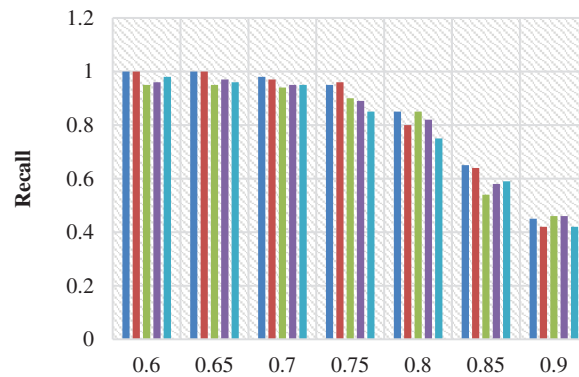


Figure 3: Recall

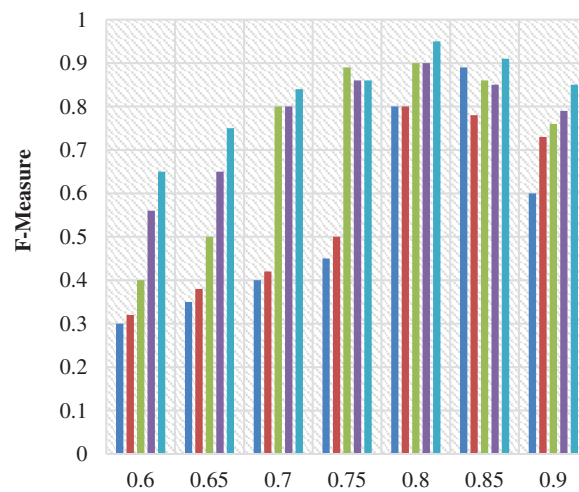


Figure 4: F-measure

Fig. 5 shows the comparison result of the proposed query recommendation with existing systems. This represents the performance result of the proposed recommendation system for the dataset of Data recommendation-Web KB. From the analysis, it is observed that the proposed service recommendation mechanism provides better results by providing high ranking services to the web pages. Also, in Fig. 6, the comparison was prepared for the accuracy and error rate of the proposed ontology based Data recommendation process.

Table 1: Allocation of documents for each class in web KB

Class	# Training	# Testing	# Total documents
Course	866	44	930
Faculty	1090	34	1124
Project	484	20	504
Student	1513	128	1641
Total	3973	226	4199

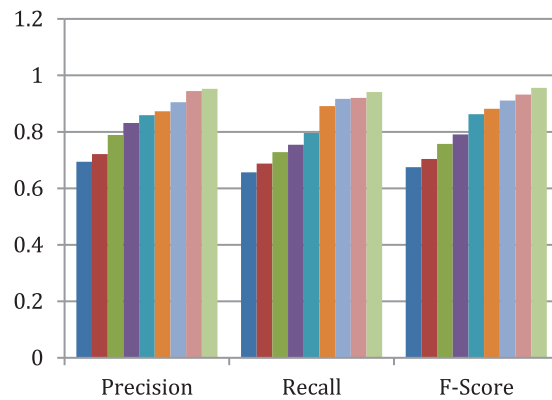


Figure 5: Overall comparison chart

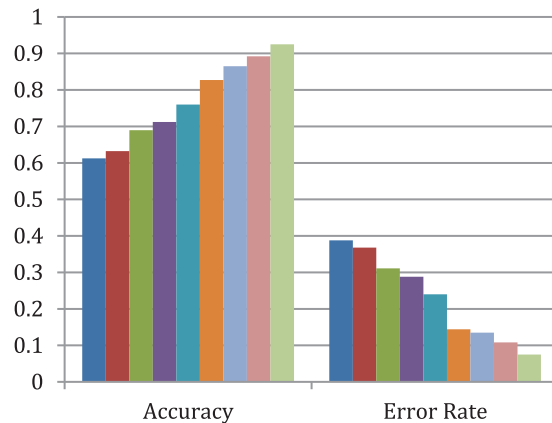


Figure 6: Comparison chart of accuracy and error rate

5 Conclusion and Future Enhancement

This paper aims to design a new query recommendation system for providing efficient services to web pages. In this, the Data recommendation based query recommendation system was focused to analyze and predict relevant data from the database. For this reason, enhanced clustering, distance based similarity, and retrieving mechanisms are utilized. Here, the stop words removal and stemming are performed to preprocess the dataset. Then, the LDTP measure is used to select the CP by extracting the keywords from the repository of files. Also, the MPS mechanism is used to form the cluster with a set of documents based on the CP. The documents are then processed for the training model in the data learning system by

using the DRNN classifier. In this environment, the server identifies the query request by using the same DRNN classification model. Moreover, the KC and TD based similarity measures are used to find the most similar items for recommendation. Also, the rank is provided for the highly similar items based on this priority by using the combination of LDTP and MPS similarity estimation technique. In performance evaluation, different measures are used to analyze the results of the existing and proposed techniques. From the evaluation, it is observed that the proposed query recommendation system provides better results by efficiently ranking the services.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] X. Chen, Z. Zheng, X. Liu, Z. Huang and H. sun, "Personalized QoS-aware web service recommendation and visualization," *IEEE Transactions on Services Computing*, vol. 6, no. 1, pp. 35–47, 2011.
- [2] X. Chen, Z. Zheng, Q. Yu and M. R. Lyu, "Web service recommendation via exploiting location and QoS information," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1913–1924, 2013.
- [3] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu and F. Yang, "Reputation measurement and malicious feedback rating prevention in web service recommendation systems," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 755–767, 2014.
- [4] P. He, J. Zhu, Z. Zheng, J. Xu and M. R. Lyu, "Location-based hierarchical matrix factorization for web service recommendation," in *2014 IEEE Int. Conf. on Web Services*, Anchorage, AK, USA, pp. 297–304, 2014.
- [5] C. Wu, W. Qiu, Z. Zheng, X. Wang and X. Yang, "Qos prediction of web services based on two-phase K-means clustering," in *2015 IEEE Int. Conf. on Web Services*, New York, NY, USA, pp. 161–168, 2015.
- [6] S. Agarwal and S. Mehta, "Approximate shortest distance computing using k-medoids clustering," *Annals of Data Science*, vol. 4, no. 4, pp. 547–564, 2017.
- [7] L. Lemos, F. Daniel and B. Benatallah, "Web service composition: A survey of techniques and tools," *ACM Computing Surveys*, vol. 48, no. 3, pp. 1–41, 2015.
- [8] J. Wu, L. Chen, Z. Zheng, M. R. Lyu and Z. Wu, "Clustering web services to facilitate service discovery," *Knowledge and Information Systems*, vol. 38, no. 1, pp. 207–229, 2014.
- [9] Z. Fu, X. Sun, N. Linge and L. Zhou, "Achieving effective cloud search services: Multi-keyword ranked search over encrypted cloud data supporting synonym query," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 164–172, 2014.
- [10] J. Lu, D. Wu, M. Mao, W. Wang and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [11] J. Liu, M. Tang, Z. Zheng, X. Liu L and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 686–699, 2015.
- [12] T. Schnabel, P. N. Bennett, S. T. Dumais and T. Joachims, "Using shortlists to support decision making and improve recommender system performance," in *Proc. of the 25th Int. Conf. on World Wide Web*, Montreal, Quebec, Canada, pp. 987–997, 2016.
- [13] Y. Xu, J. Yin, S. Deng, N. N. Xiong and J. Huang, "Context-aware QoS prediction for web service recommendation and selection," *Expert Systems with Applications*, vol. 53, pp. 75–86, 2016.
- [14] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453–466, 2015.
- [15] G. Kang, M. Tang, J. Liu, X. Liu and B. Cao, "Diversifying web service recommendation results via exploring service usage history," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 566–579, 2016.

- [16] B. Bouihi and M. Bahaj, "An ontology-based architecture for context recommendation system in e-learning and mobile-learning applications," in *2017 Int. Conf. on Electrical and Information Technologies*, Rabat, Morocco, pp. 1–6, 2017.
- [17] Z. Gulzar and A. A. Leema, "Course recommendation based on query classification approach," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 13, no. 3, pp. 69–83, 2018.
- [18] J. K. Tarus, Z. Niu and G. Mustafa, "Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 21–48, 2018.
- [19] G. George and A. M. Lal, "Review of ontology-based recommender systems in e-learning," *Computers & Education*, vol. 142, pp. 103642, 2019.
- [20] D. Shi, T. Wang, H. Xing and H. Xu, "A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning," *Knowledge-Based Systems*, vol. 195, pp. 105618, 2020.
- [21] C. D. Medio, C. Limongelli, F. Sciarrone and M. Temperini, "MoodleREC: A recommendation system for creating courses using the moodle e-learning platform," *Computers in Human Behavior*, vol. 104, pp. 106168, 2020.
- [22] X. Zhang, W. Zhang, W. Sun, X. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
- [23] X. Zhang, X. Sun, X. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
- [24] F. Chen, C. Lu, H. Wu and M. Li, "A semantic similarity measure integrating multiple conceptual relationships for web service discovery," *Expert Systems with Applications*, vol. 67, pp. 19–31, 2017.
- [25] T. S. Ibrahim, A. I. Saleh, N. Elgaml and M. M. Abdelsalam, "A fog based recommendation system for promoting the performance of learning environments," *Computers & Electrical Engineering*, vol. 87, pp. 106791, 2020.