Design of Query Processing System to Retrieve Information from Social Network using NLP

Charu Virmani[1], Dimple Juneja[2], Anuradha Pillai[3*]
[1]Research Scholar
YMCA University of Science and Technology
Faridabad, 121006, INDIA
E-mail:charu.fet@mriu.edu.in
[2]Department of Computer Applications,
National Institute of Technology
Kurukshetra, 136119, INDIA
E-mail:dimplejunejagupta@gmail.com
[3]Department of Computer Engineering,
YMCA University of Science and Technology
Faridabad, 121006, INDIA
*Corresponding author: Anuradha Pillai

Received May 18, 2017; revised August 20, 2017; revised September 10, 2017; revised September 28, 2017;accepted October 11, 2017; published March 31, 2018

Abstract

Social Network Aggregators are used to maintain and manage manifold accounts over multiple online social networks. Displaying the Activity feed for each social network on a common dashboard has been the status quo of social aggregators for long, however retrieving the desired data from various social networks is a major concern. A user inputs the query desiring the specific outcome from the social networks. Since the intention of the query is solely known by user, therefore the output of the query may not be as per user's expectation unless the system considers 'user-centric' factors. Moreover, the quality of solution depends on these user-centric factors, the user inclination and the nature of the network as well. Thus, there is a need for a system that understands the user's intent serving structured objects. Further, choosing the best execution and optimal ranking functions is also a high priority concern. The current work finds motivation from the above requirements and thus proposes the design of a query processing system to retrieve information from social network that extracts user's intent from various social networks. For further improvements in the research the machine learning techniques are incorporated such as Latent Dirichlet Algorithm (LDA) and Ranking Algorithm to improve the query results and fetch the information using data mining techniques.The proposed framework uniquely contributes a user-centric query retrieval model based on natural language and it is worth mentioning that the proposed framework is efficient when compared on temporal metrics. The proposed Query Processing System to Retrieve Information from Social Network (QPSSN) will increase the discoverability of the user, helps the businesses to collaboratively execute promotions, determine new networks and people. It is an innovative approach to investigate the new

http://doi.org/10.3837/tiis.2018.03.011 **ISSN : 1976-7277**

aspects of social network. The proposed model offers a significant breakthrough scoring up to precision and recall respectively.

Keywords: Query Processing, Social Network, LDA, Natural Language Processing

1. Introduction

Query processing in Social Network Aggregators (SNA) [1] extracts user information from multiple social networks in a reliable way. Conventionally, the respective query engines of the social network respond to the user's request by using conventional information retrieval methods [1] which in turn return a huge lot comprising of both relevant and irrelevant information. The proposed method offers an edge over other mechanisms as it not only retrieves more user-centric results as compared to traditional way of keyword-based searching but also in timely manner as well. The system exploits natural language techniques for query processing and extracting information from the social web. Although natural language techniques are finding space in semantic search engines, however; the same has not been applied to respond to queries executed on social networks. Thus the motivation for the proposed model is to find a viable solution that can provide an intelligent and integrated result of user's free form query.

The study of the literature reveals various works and application of information systems in social networks. Matsuo et. al. [2] have developed a system "POLYPHONET" to extract the information from the social network that detects relationships of person, groups of person and obtain keywords for a person. In the environment of semantic web, social networks and semantics are the dualistic sides of the coin as pointed by Mika [3]. There exist several ways such as relation extraction, event detection etc. to extract the information from the social network [4][5][6]. Tyler et al. [4] has explored the detection of relations on the basis of information. Kautz [6] developed a Referral Web by extricating the measurement of the co-event of the names on the web.

In the past, the enhanced development of online social Networks (OSNs) derives the dispersion of a large amount of profile information inside corresponding social networks. Thus, sharing and reusing user's information accessible crosswise over OSNs is an emerging challenge. Xuan et al. [7] have presented a primary social user aggregation based on the FOAF ontology. However, the model has neither kept trace of the provenance nor adding time of any information. Moreover, there may be conflicting values for a given property and it is left to the user to decide if information should be kept or deleted.

Carmagnola et al. [8] have proposed a mechanized coordinating calculation which under the set of user characteristics like sexual orientation, birthday, city, can register the likely comparability between these starting characteristics and creep information from social sites. More is the information crept, more precise is the calculation. In any case, ambiguous information is freely accessible because of the closeness of the majority of prominent OSNs. Another OSN Aggregator proposed by Zhang et al. [9] not only pulls the social information from multiple networks, but also group, rate and notifies about the activities of friends.

However, the system failed to integrate the networks. In fact, numerous models have been advanced to outline a collective objective model for assimilating a user [10]. Abel et al. [11] aggregated user profiles on the limited set of properties like name, photos etc. using the most

popular solution FOAF from online social networks by applying rules.

Orlandi et al. [12] models the user interest by combining the profile information and semantic web using FOAF ontology and DBpedia resources. Shapira et al. [13] developed a collective recommender system by exploiting user inclinations. Rohani et al. developed a robust recommender system for academic social networks to recommend and analyze the products that meets the user's preferences [14]. Further to model the interest of the user, various weighted concepts using semantic web are listed in [15]. Twitter and Facebook were the domains used for extracting information for the exploration of text. Few other models have considered expertise and relationship of user into consideration and developed a social search engine [15].

Wang and Jin [16] personalized the search results by incorporating users interest from multiple social netwroks using social activities of user such as bookmarking. Zhou et al. [17] have retreived the information about the user using historical usage information of the user. Yang S. [18] have employed the collection of operators on the graph and explored the Social Network Graph Query Language for performing a search on social media in a natural language. The system was imlplemented by building database management system from scratch that can make the contol of components easy rather than taking inputs from existing social networks. Groh and Hauffa [19] have characterized the social relationships using unsupervised learning and natural language techniques for the purpose of linguistic analysis on the classification of sentiment polarity.

Shojafar et al. [20] explored two novel parameter tunable frameworks for efficient route discovery with the use of topological information and collaborating peers. A uniform watchtower framework for delay efficient with minimum bandwidth cost and latency to discover the route was proposed. Cordeschi et al. [21] described a scalable scheduler for optimizing the streams of Big Data, which takes the advantage of optimal distribution of the virtual resources. Mukhopadhyay et al. [22] proposed research study works effectively and handles the challenges of relevant not reachable web pages.

There have been numerous attempts to search for the structured data but none of the attempts is appropriate for unstructured data search engine [23]. However, structured database supports text indexing but they agonize from poor performance [12][13]. The dwelling of literature clearly indicates the fact that aggregating the profiles of the social network provides a large amount of information about the user and no author has made an attempt to extract the information from this pool in a user-friendly way. The work proposes the architecture of retrieving the information from the query written in a natural language for the social network aggregator by extracting entity, mapping it to its semantic meaning, identifying user preferred profiles and improving upon the user's preference by ranking the profiles.

This paper is structured into five sections: Section 1 introduces and throws light on the work of eminent researchers highlighting the substantial contributions. The discussion in section 1 illustrates that none of the authors till date has tried to extract the information from the social network by processing the query in a natural language. The current work thus finds motivation and resolves the challenge listed above. Section 2 uniquely contributes a Query processing System (QPS), a content Based Semantic Match Maker (CBSMM), a Machine Learning mechanism (MLM) and a Ranking mechanism. In contrast to QPS which extracts the entities of a query, CBSMM performs matching of these entities to their semantic meaning. MLM extracts the identities from the social network and Ranking sorts the output user's profiles as per the query and other prevailing factors. Section 3 details the case study of the proposed work. This has been established with a data set in the results and discussion section given in

section 4. Section 5 finally concludes.

2. Proposed Work

Consider a scenario where a user A wish to search for a Female and knows Java. In order to address such a query, in a conventional mode, user A would explore the popular social networks adding the additional details such as work sector. Integrating such profiles is a challenge for naive users and a simple solution to address the issue is highly desired. The section proposes a technique for smooth management of extracting information from user profile across multiple social networks.

The proposed model provides a natural way of managing, processing, and analyzing the complex, heterogeneous unstructured data. Designing such a new system that accommodates the voluminous data requires rethinking all aspects of a DBMS, including data modeling, storage management, indexing, query processing and optimization. It shall allow the entire social web to give personalized content or recommendation to the formal queries. QPSSN extracts user's public information and preferences across their online presence. The result of the intelligent search is the direct answer to the user's query instead of the networks to follow. The user-centric search shall help users find places, skills, users or product that their friends or other people in the network have. It will improve the discoverability of a user in the social network for businesses and implication for many companies. As outlined in **Fig. 1**, the proposed model primarily comprises of four modules namely, Query Processing System (QPS), Content Based Semantic Matcher Maker (CBSMM), Machine Learning Mechanism (MLM) and Ranking of results.

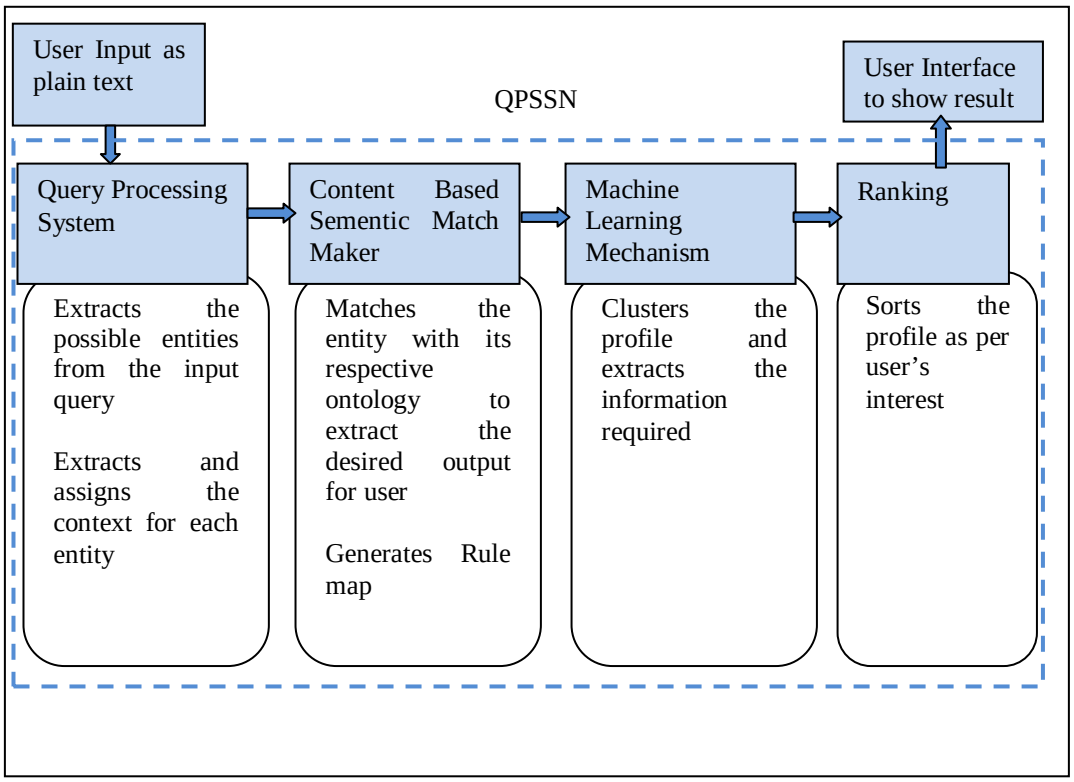

Fig. 1. Architeture of QPSSN

2.1 Query Processing System (QPS)

Primary aim of QPS is to extract the possible entities from the input query and determine the context of each entity. It exploits existing NLP techniques [24]. QPS majorly contributes towards entity recognition (noun phrase) and its extraction and executes in four phases namely user interface, preprocessing, entity tagging and context extractor (see **Fig. 2**).

QPS takes user query as inputs form user interface which in turn are pre-processed using parsers and stemmers generating entities. During pre-processing, query text is tokenized and cleaned (the determiners are characterized under stop words) by removing all stop words. This module finds its space from the list of stop rundown of words which are insignificant for the input [25]. QPS makes use of Morphological analyzer [26] and Porter's stemming algorithm [27] to establish relation between the words and stemming of the words to its root.

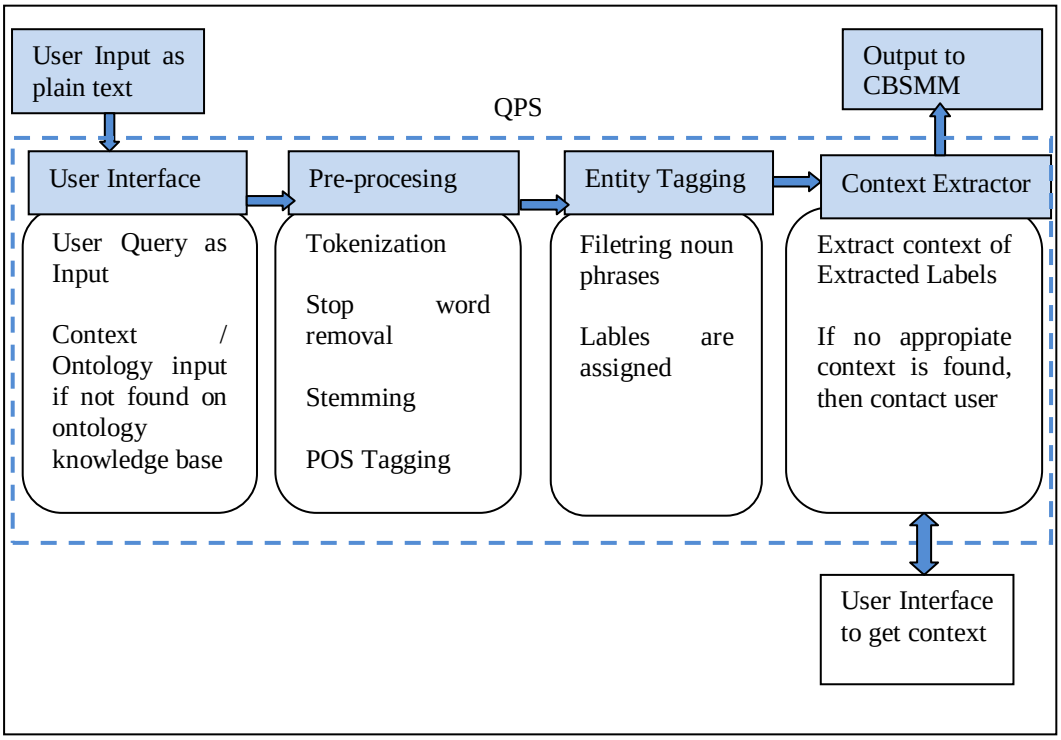

Fig. 2. QPS Pipeline

In the next phase i.e. Entity Tagging, labels are assigned to the connected words produced during previous phase. An entity tag differentiates the word as noun, pronouns, descriptors, determiners and verb. Entity tagging is based on Penn Tree Bank Parser [28] that interprets structure of the phrase. Entity tagging results into tagged entities that serves as input to the context extractor which in turn returns the ontology of the entities. The Context extractor tries to find a matching Ontology from the Ontology knowledgebase. In case no matching Ontology could be found, the context extractor requests the desired ontology from the user and in turn generates semantic information about the tagged entities. This is achieved by a user interface which does the required integration with user to get the required context. It is worth

mentioning that context extractor of QPS is assisted with an ontology knowledge base containing ontological data to generate pragmatics. Algorithm for QPS is as shown in Algorithm 1. The semantic information thus generated serves as input to CBSMM to enhance the semantic ontological information. The mapping of context given by QPS and enhanced context by CBSMM results into the resolution of heterogeneity in ontology.

```
Algorithm1 : Query Processing System

Input: Query in Natural Language
Output: Context

QPS(Query)
{
cleaned_Query = Activate PreProcessing(Query);
tagged_Entity = Activate EntityTagging(cleaned_query);
Context = Activate ContextExtractor(tagged_Entity)
return Context;
}

PreProcessing(Query)
{
Nonword = Identify nonword(query)
If nonword ≠ NULL
    delete nonword tokens
Words = Parse(text)
stopword = Identify stopwords(words)
If stopword ≠ NULL
       Remove stopwords
cleaned_Query = Stemming (!stopwords)//Porter's stemming algorithm
Return cleaned_Query
}

EntityTagging (cleaned_query)
{
For each keyword from cleaned_query
      tagged_Entity = tag(keyword) // use Penn Tree Bank Parser
Return tagged_Entity
}

ContextExtractor(tagged_Entity)
{
For each tagged_Entity
      Context = search ontologyknowledgebase(tagged_Entity)
     If context ≠ NULL   then
       Return context
    Else
       Context  =  user_interface(tagged_Entity)
      Return Context
}
```

2.2 Context Based Semantic Match Maker (CBSMM)

CBSMM refines the context obtained from the QPS to enhance and enrich the context. This essentially means that the scope of the context obtained will be intelligently increased so that more relevant and wider knowledge is retrieved. The domain knowledge is fed by an in-build knowledge base with learning capability. CBSMM follow the following rule map to augment the context.

1) Context Engine analyzes the context obtained from QLP module and fetches the relevant ontologies from the knowledge base. This is called context enrichment and enhancement because the scope of context will now be improved to include other relevant ontologies as well
2) Knowledge base is a database where all context and ontologies which the system possess or have learnt are stored.
3) Rule Mapping will map the entities to provide all semantically related terms and context which will eventually increase the scope of the context search. This rule map will serve as the input to next module MLM
4) Learning engine will feed knowledge into the knowledge base for every new context-ontology pair. Possibly obtained from the user inputs in QLP module.

A simple flow of information in CBSMM is shown in **Fig. 3** and algorithm is shown in Algorithm 2.

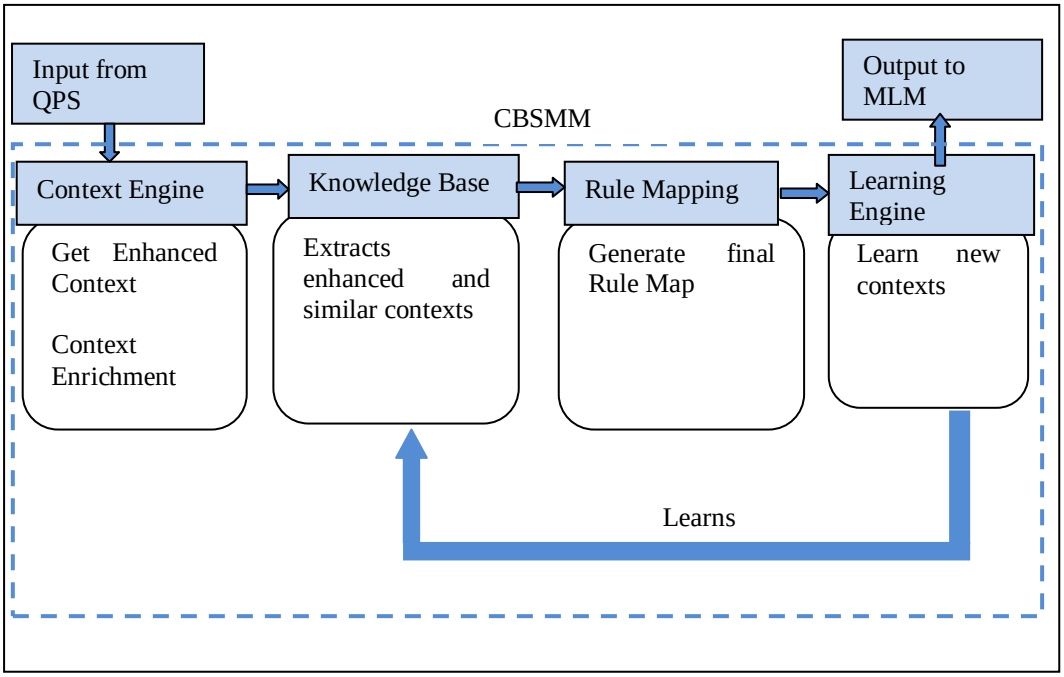

Fig. 3. CBSMM Pipeline

```
Algorithm 2:  Context Based Semantic Match Maker
    Input: Context
    Output: Rule_Ontology

    CBSMM(Context)
    {
    Rule_ontology= Activate Context Engine(Context);
    Return Rule_ontology
    }

    Context Engine(context)
    {
    semantic_info = Search KnowledgeBase(context);
    if semantic_info ≠ NULL then
    for each context in semantic_info
    Rule_ontology = map ContextRuleMap(context,,semantic_info);
    Return Rule_ontology
   }
```

2.3 Machine Learning Mechanism

MLM is the main module that returns the search results based on the refined ontologies as processed by CBSMM Module. Each context is searched in cluster registry to obtain appropriate matching cluster(s) that matched user criteria. In case there is no appropriate cluster available, the MLM dynamically generates the cluster. The architecture of MLM is elaborated in **Fig. 4** and algorithm in Algorithm 3.

MLM has four components that return the set of user profiles. The following states the process of MLM:-

1) CRegistry: It maps the context to cluster id (CID) that is related to a particular context. This CID is then used to fetch the information from the CDatabase.
2) CEngine: The CEngine is associated with CDatabase and invokes particular cluster from the CDatabase on the basis of CID given by CRegistry. It also entreats the CCreation if a relevant CID is not returned by the CRegistry. The cluster generated from the CCreation is then stored in the CDatabase, an id is allocated and stored into CRegistry.
3) CDatabase: It consists of clusters of user profiles having useful information about the user. It also manages, synchronizes and collaborates with each other to find the composite solution when an optimum CID is not returned by the CRegistry.
4) CCreation: It create the clusters dynamically to fetch the relevant data, if the CDatabase does not have a cluster corresponding to the ontology, cluster is created and an entry is also made to CDatabase and CRegistry.

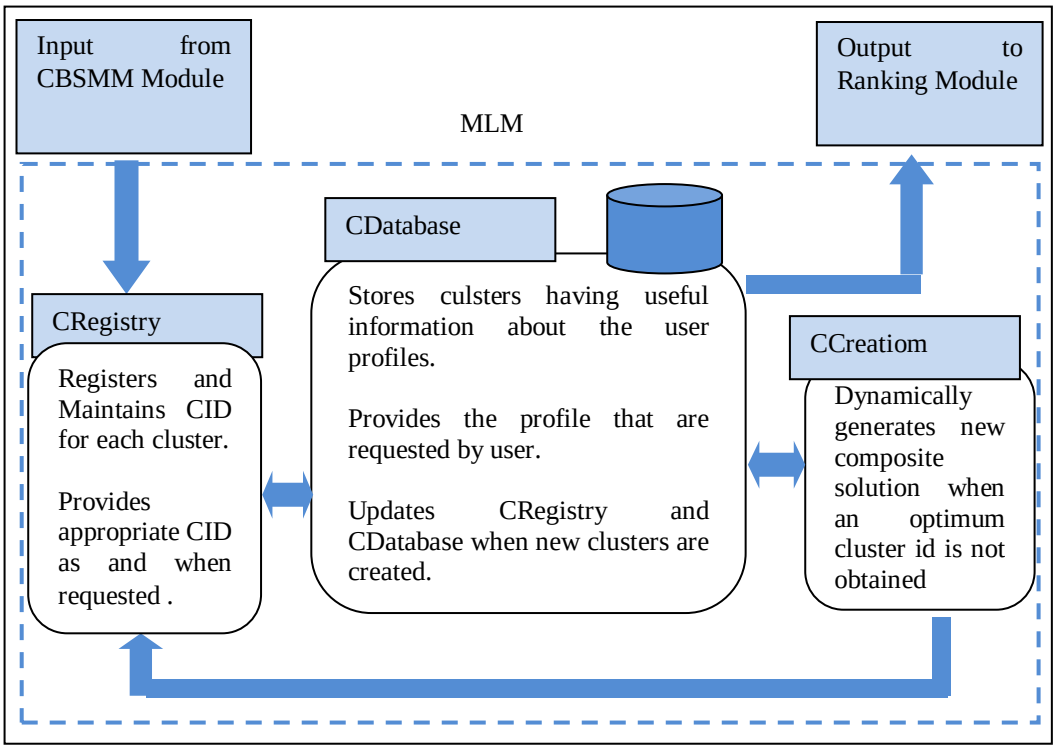

Fig. 4. MLM Architecture

CBSMM employs LDA model [29] to cluster the documents into clusters. A cluster optimum id (CID) is returned to the CBSMM when a request is generated. This id is then responsible to find an appropriate cluster for the user input entities from the CDatabase, however if the system fails to find an optimum cluster then it generates a special request to CCreation to compose a new solution that exploits entities from the previous known clusters and then the new solution is saved in the CDatabase with a new id and is kept for future purposes. The composite solution provides the relevant results close to the query but may or may not be exactly as requested by the user. It registers itself with CRegistry and a unique id is allocated to it.

It is an obvious fact from the above discussion that the MLM has the ability to learn and act pro-actively in the environment and produce better results. It increases the probability of success rate by dynamically adding composite clusters when an appropriate cluster is not found by the MLM. Considering the queries of all users enrolled on social network, it always return the user's public profile information like name, photo, location etc. independent of the query. Preparing the query once and reusing it, ensures that planning is done once and hence increase the overall performance. The algorithm for MLM is shown in algorithm 3 and algorithm 3(a) depicts the pseudocode for generating dynamic clusters.

```
Algorithm 3 : MLM
Input: Rule_ontology,query
Output: Expected_Users

MLM(Rule_ontology,query)
  {
```

```
   CID=search CRegistry(Rule_ontology)
      If CID ≠ NULL then
        Cluster_user_profiles = search CDatabase(CID)
         For each user = user1 in cluster_user_profiles
                                  Accuracy_Score = calculate accuracy(user1,query);
                                 Threshold_score=0.85;
                     If (threshold score>Accuracy_Score);
                  Return user1;
      Else
            Dynamic_solution = Activate Dynamic_Cluster(users,query);
            Return Dynamic_solution_user1;
}
```

```
Algorithm 3a  : Generate_Dynamic_Clusters
Input: User_Profiles
Output: Clusters
Dynamic_Cluster (users,query)
          {
          Clusters = Activate LDA_Clusters(users,query);
          Clusters_info = Extract Info(Clusters);
          Register Clusters and Clusters_info in CDatabase and assign a CID;
          Register CID in CRegistry
          If CID ≠ NULL then
                    For each Dynamic_Solution_user = user1 in Clusters;
                              Accuracy_Score = calculate accuracy(user1,query);
                             Threshold_score=0.85;
                    If (threshold score>Accuracy_Score);
                  Return Dynamic_Solution_user1;

          }
```

2.4 Ranking

This module ranks the user profiles resulting from the MLM on the basis of weighted score so that most relevant profile results first. The Rank of the user profile is computed by a weighted score of the user to the group. The ranking architecture encompasses two major components to rank the desired documents i.e. Rank Engine and Sorting as shown in **Fig. 5** and the algorithm is shown in algorithm 4.

The Rank Engine computes the Rank or weighted score based on user to user interaction. The user interaction could be further categorized as user-group, user-user and mutual friends. Here, User-Group Interaction determines the common groups between the searcher and users from the seed. If a common group exists, then the user who frequently interacts and has recently interacted with the group will be more important than the one who least interacts. User-User Interaction determines the score of interaction between two users. The recent interaction between the searcher and the user returened by the cluster will be given a high edge over another. In case of mutual friends, the score of common friends between the two user is evaluated. The sorting module sorts the user profile on the basis of the weighted score as calculated from above three approaches and provides the output to the user interface.

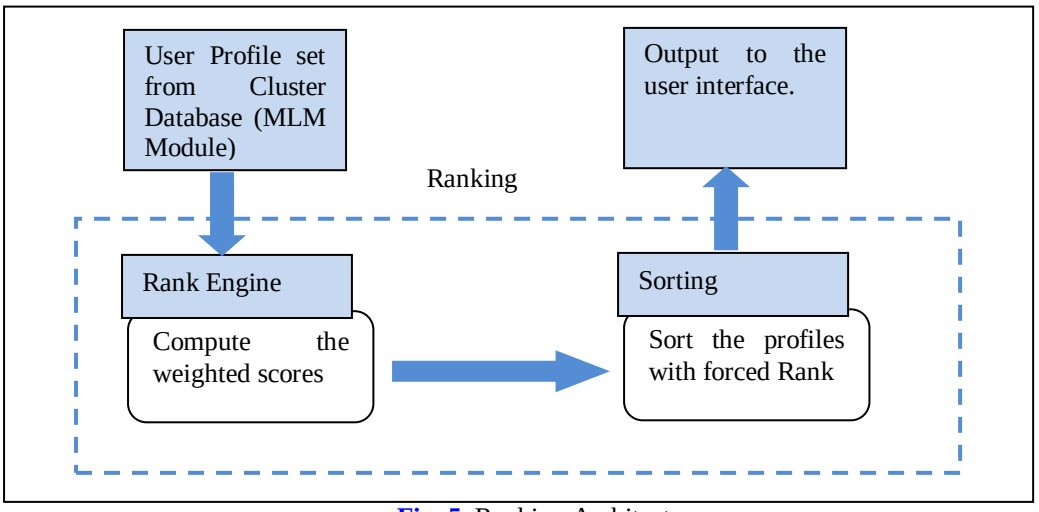

Fig. 5. Ranking Architecture

```
Algorithm 4 Ranking
Input: Expected_Users
Output: Sorted profiles

Rank(Expected_Users)
        {
          user1 = user1_info();  // the user who input the query
For each user= user2 in Expected_Users  c.C
           User_Interaction = Activate User_User_Interaction(user1,user2);
          Group = Search_Common_Group(user1,user2)
          For each Group.i in Group
                    Group_Interaction = Activate Group_User_Interaction(Group,user1,user2);
          Group_Interaction_Score=(Group_Interaction1 + Group_Interaction2 + ---- ---- --- ---
+ _Group_Interaction n)/n
          Mutual_friends = Activate Mutual_Friends(user1,user2);
          Weighted_score=(User_Interaction+Group_Interaction_score+Mutual_Friends)/3;
           Sorted_Profiles = Sort(weighted_score);
Return Sorted_Profiles;
      }

 User_User_Interaction(user1,user2)
{
For each post or tweets in user1
If  Likes(user2) or follows(user2) then
        User_Interaction =User_Interaction+1
For each post or tweets in user2
If  Likes(user1) or follows(user1) then
        User_Interaction =User_Interaction+1
Return User_Interaction
}
```

```
  Group_User_Interaction(Group,user1,user2)
{
For each Group in user1 and user2
      For each post in Group
            If  Likes(user1) and Likes(user2) then
                  Group_User_Interaction =Group_User_Interaction+1
      Return Group_Uer_Interaction
}

  Mutual_Friends(user1,user2)
{
For each SN user1 belongs
            score=Extract_Common_Friends(user1,user2)
Mutual_Friends=(score1+score2___+scoren)/n
Return Mutual_Friends
}
```

The next section illustrates the working of QPSSN with a case study implemented on Intel Core i5 with 8GB RAM using Windows 7 Operating System.

3 The Case Study

The algorithm was given the input query “Friends who knows java” and the description of each step by step output thus obtained is as illustrated below.

Input: Friends who know Java.
Output: Sorted User Profile set

- **Processing in QPS Module**

Input: Friends who Know Java

Following steps were followed
1) The input string in tokenized into keywords: Friends, who, know, Java
2) Stemming is performed to remove the filler words. Resultant keywords are Friends, Java
3) Keywords are tagged as Noun
4) Friends is assumed to be a known keyword and Java being an ambiguous keyword is tagged to Skill, Location, Name ontologies
5) User was asked to suggest ontology for Java; and user chose the ontology Skill.

Output: keyword-ontology Pair as shown in **Table 1.**

Table 1. Keyword-Ontology Pair

Keyword	**Ontology**	**User Input**
Friends	Friends	N
Java	Skill	Y

- ***Processing in CBSMM Module***

Input: **Table 1:** Keyword-Ontology Pair

Following steps were followed:

1) Context Engine takes the relevant Ontologies as input.
2) Context Engine Searches the refined Ontologies from the Context Knowledge base and results some related ontologies.
3) Rule map has created which maps the keyword to other Ontologies. Let us assume that our rule map defines the following for the keyword Friends (language) is as shown in **Table 2.**

Table 2. Enhanced related terms

Keyword	Ontology
Friends	Friends: Mutual Friends, Tagged friends, Shared links etc.
Java as a skill	Skill: Eclipse, Core Java, Beans , EJB, Sun Microsystems

Output: Enhanced Ontology and Rule: Friends(language)

- ***Processing in MLM Module***

Input: Enhanced Ontology and Rule: Friends(language)

Following steps were followed

1) Search the CRegistry for CID matching with Ontologies and the result of CRegistry is as shown in **Table 3.**

Table 3. CID

CID	Ontology
C_Friends	Friend
C_Java	Java
C_Eclipse	Eclipse
C_Beans	Beans
C_sun	Sun microsystems

2) CEngine retrieves the relevant user profile set from the CDatabase as shown in **Table 4.**

Table 4. Output of cluster engine

Cluster ID	User Profiles
C_Friends	P1, P2
C_Java	P2, P3, P4
C_Eclipse	P2,P4
C_Beans	P2,P4
C_Sun	P2,P4,P5

Output: User Profile Set {P1, P2,P3,P4,P5}

- ***Processing in Ranking Module***

Input: User Profile Set {P1, P2,P3,P4,P5}

Following steps are performed

1) Weighted score for each user profile. Demo values shown in **Table 5**.

Table 5. Weighted score

User Profile	Weighted scores
P1	0.2
P2	0.8
P3	0.5
P4	0.7
P5	0.2

2) Sorting of user profiles based on scores {P2,P4,P3,P1,P5}

Output: Ranked User profile set {P2,P4,P3,P1,P5}

4. Results and Discussions

This section describes the experiments on the data set from multiple social networks as discussed in the previous section. The data collection system extracts different data metrics from different social media platforms with the help of input search queries.

The system has selected 20 different user-skills as the input queries such as - “java”, “python” and “mongo” etc. The system collects the different data metrics such as - user-name, user-location, user-description, gender, birth, connections, tweets, etc. from multiple social media platforms by using user’s information from the online social media API‘s. The system collected user-level data and user demographics using Twitter public search. The system collected 67,956 documents. The mixed inputs of user-variables in the Bing Search API’s are used to collect the information of around 87,734 links out of them 35,413 were user profile links, however only 26,543 of them had exact matches with the input-queries of the twitter public search. The system has also used FullContact to extract the alternative user-profiles of a twitter user. The system collected total 24,341 user-links from Fullcontact, These results were merged with those obtained from Search Engine Automation component. It is worth to mention that due to the restriction of offered attributes by API’s of different social networks, the system could only use limited features of profile. The user’s information is collected using access token and a prior approval from the users has taken to use the user’s data for the research purpose only. A sample used 100 set of queries for testing the system. The samples of input query are:

- Friends who live in U.S and knows DataScience
- Friends who are single and above 25 years
- Friends who works in TCS and is female
- People who live in London
- Looking for Java developer
- We require Java developer
- We require Java developer who lives in Delhi
- People who live in London and knows Python
- Looking for Java developer who lives in Gurgaon
- Friends who live in U.S and knows DataScience
- Friends who live in India and knows DataScience
- Friends who live in U.S and knows Database

The system utilizes Latent Dirichlet Allocation (LDA) clustering to create clusters and allocates a reference id to each cluster. The clustered data are then trained in the classifier. The entities placed to form the users are collected and fed to classifier. The classifier compares the query with the trained data and finally classified results are obtained as the output. The output of language cluster depicting the skills for particular location U.S. is shown in **Fig. 6**. **Fig. 7** represents the count of users for different cities of U.S. for the skills datascience, database and javascript.

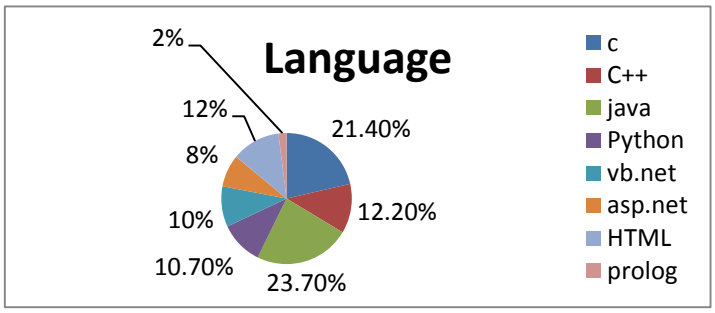

Fig. 6. Clustering Output – Language

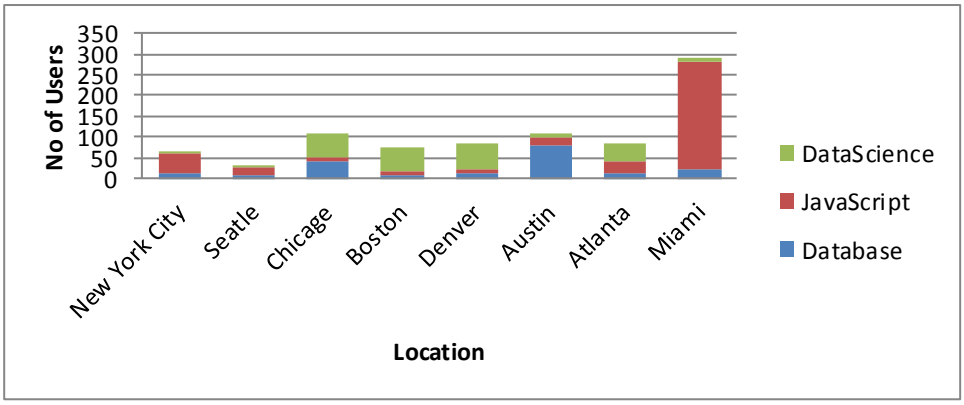

Fig. 7. Count of User for Different Skills for Different Locations

For the input queries, the system is able to extract the relevant profiles from these clusters as per the cluster id**.** During the course of implementation, it has been observed that the system results in the expected user profile at the top 3 ranks. The query processing dialog box is as shown below **Fig. 8.** Potential user profiles available from multiple social networks has recommended according to the given query. The query given is the search of java developer from various social network user profiles. The Output of the query is as shown in **Fig. 9**.

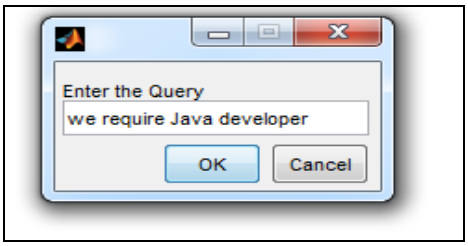

Fig. 8. Query processing Dialog box

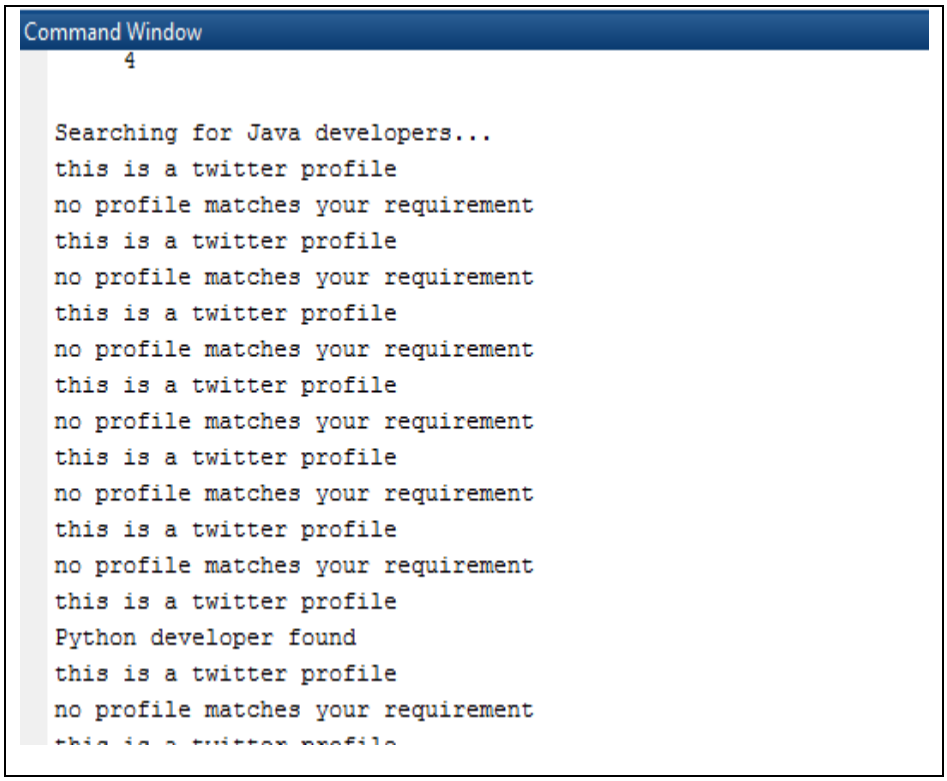

Fig. 9. Output of Matlab

In information retrieval and query processing, the precision is the fraction of retrieved documents that are relevant to the query.

Precision = Total Number of Relevant user profiles found / (Total Number of Relevant user profiles found + Total Number of irrelevant user profiles)

High precision depicts that the quality of retrieved results achieves the performance close to the expectations of the users. The proposed work was carried on 1000 number of user profiles to 5000 users profiles. The precision value, true positive rate and false positive rate of the query search are obtained. The Precision of QPPSN to measure the accuracy and the traditional keyword search is compared in **Fig. 10**.

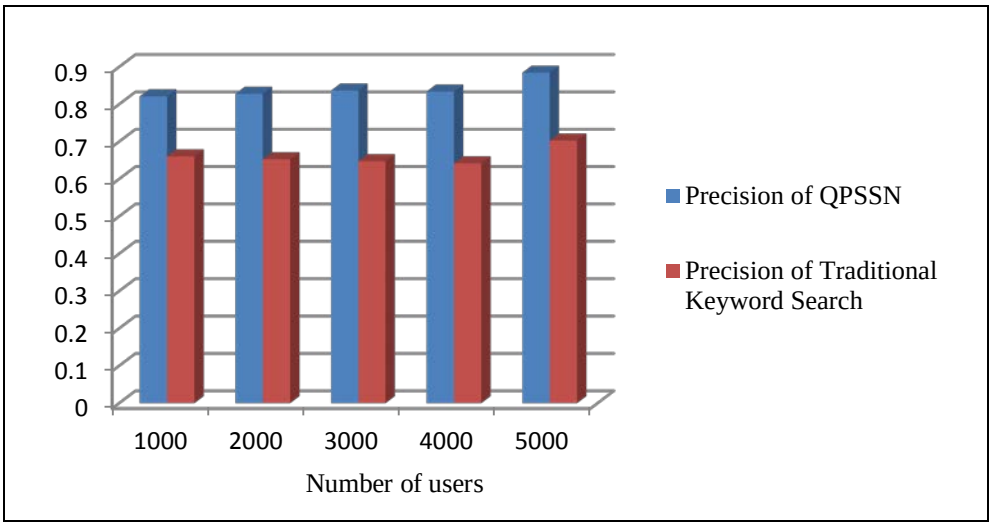

Fig. 10. Comparison of Precision Value to Measure Accuracy

A Receiver Operating Characteristic (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on its performance. It represents a relationship between sensitivity (Recall) and specificity. It is a tool to evaluate quality of cluster production, which shows the actual positive rate on the Y-axis and the curve showing the false positive rate on the X-axis. True positive means a match output agrees with the value present in the test case. The threshold is set to be limited, resulting in a false positive rate of less than 4%. The ROC curve of the query is as shown in **Fig. 11**.

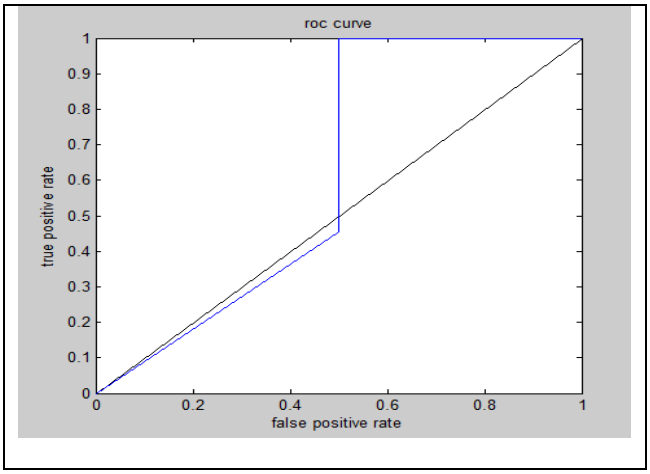

Fig. 11. Roc Curve Depicting the True Positive Rate Vs False Positive Rate for Test Corpus

The result of the query is the first best option that meets the criteria as given in **Fig. 12**

Fig. 12. Result of the Query

The sorted list of output is as shown in **Fig. 13.**

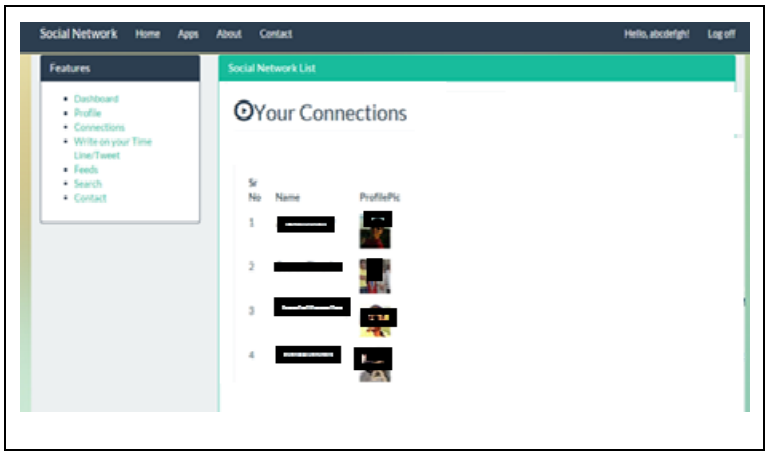

Fig. 13. Sorted List of Query results

5 Conclusions

QPSSN proposed to extract the information from the user profile from various social networks using an efficient algorithm that takes the input query in a natural language. The system is tested for a set of 100 queries and derived that the expected result results in top 3 users by the system. The simulated results have shown the fact that natural language processing on the query using multiple social networks will increase the discoverability of the user, will help the organizations and businesses to collaboratively execute promotions, determine new networks and people. It is more efficient and superior to the advanced user profile mapping using keyword searching. The proposed research method can be used for background verification purpose, recruitment agencies, targeting specific group of people. However, the system only considered the public available attributes, in the future; it can be extended to search for post/tweets to make provision for real-time interaction.

References

[1] Virmani, C., Pillai, A., Juneja, D., "Study and analysis of Social network Aggregator," in *Proc. of Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, IEEE*, pp. 145-148, 2014. Article (CrossRef Link)

[2] Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K., Ishizuka, M., "POLYPHONET: an advanced social network extraction system from the web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 4, pp. 262-278, 2007. Article (CrossRef Link)

[3] Mika, P., "Ontologies are us: A unified model of social networks and semantics," *Web semantics: science, services and agents on the World Wide Web*, vol. 5, no. 1, pp. 5-15, 2007. Article (CrossRef Link)

[4] Tyler, J. R., Wilkinson, D. M., Huberman, B. A., "E-mail as spectroscopy: Automated discovery of community structure within organizations," *The Information Society*, vol. 21, no. 2, pp. 143-153, 2005. Article (CrossRef Link)

[5] Miki, T., Nomura, T., Ishida, T., "Semantic web link analysis to discover social relationships in academic communities," in *Proc. of Applications and the Internet, 2005. Proceedings. The 2005 Symposium on, IEEE*, pp. 38-45, 2005. Article (CrossRef Link)

[6] Kautz, H., Selman, B., Shah, M., "The hidden web," *AI magazine*, vol. 18, no. 2, pp. 27, 1997. Article (CrossRef Link)

[7] Vu, X. T., Morizet-Mahoudeaux, P., Abel, M. H., "User-centered social network profiles integration," in *Proc. of 9th International Conference on Web Information Systems and Technologies, SciTePress*, pp. 473-476, 2013. Article (CrossRef Link)

[8] Carmagnola, F., Osborne, F., Torre, I., "User data distributed on the social web: how to identify users on different social systems and collecting data about them," in *Proc. of Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, ACM*, pp. 9-15, 2010. Article (CrossRef Link)

[9] Zhang, J., Wang, Y., Vassileva, J., "SocConnect: A personalized social network aggregator and recommender," *Information Processing & Management*, vol. 49, no. 3, pp. 721-737, 2013. Article (CrossRef Link)

[10] Carmagnola, F., Cena, F., Gena, C., "User model interoperability: a survey," *User Modeling and User-Adapted Interaction*, vol. 21, no. 3, pp. 285-331, 2011. Article (CrossRef Link)

[11] Abel, F., Henze, N., Herder, E., Krause, D., "Linkage, aggregation, alignment and enrichment of public user profiles with Mypes," in *Proc. of Proceedings of the 6th International Conference on Semantic Systems, ACM*, p. 11, 2010. Article (CrossRef Link)

[12] Orlandi, F., Breslin, J., Passant, "Aggregated, interoperable and multi-domain user profiles for the social web," in *Proc. of Proceedings of the 8th International Conference on Semantic Systems*, ACM, pp. 41-48, 2012. Article (CrossRef Link)

[13] Shapira, B., Rokach, L., Freilikhman, S., "Facebook single and cross domain data for recommendation systems," *User Modeling and User-Adapted Interaction*, pp. 1-37, 2013. Article (CrossRef Link)

[14] V. A., Kasirun, Z. M., Kumar, S., Shamshirband, S., "An effective recommender algorithm for cold-start problem in academic social networks," *Mathematical Problems in Engineering*, 2014. Article (CrossRef Link)

[15] Zhou, M., Zhang, W., Smith, B., Varga, E., Farias, M., Badenes, H., "Finding someone in my social directory whom i do not fully remember or barely know," in *Proc. of Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, ACM*, pp. 203-206, 2012 Article (CrossRef Link)

[16] Wang, Q., Jin, H., "Exploring online social activities for adaptive search personalization," in *Proc. of Proceedings of the 19th ACM international conference on Information and knowledge management, ACM*, pp. 999-1008, 2010. Article (CrossRef Link)

[17] Zhou, D., Lawless, S., Wu, X., Zhao, W., Liu, J., Lewandowski, D., 'A study of user profile representation for personalized cross-language information retrieval," *Aslib Journal of Information Management*, vol. 68, no. 4, 2016. Article (CrossRef Link)
[18] YANG, S., "Data Modeling and Query Processing for Online Social Networking Services (Doctoral dissertation)", 2011. Article (CrossRef Link)
[19] Groh, G., Hauffa, J., "Characterizing Social Relations Via NLP-Based Sentiment Analysis," in *Proc. of ICWSM*, 2011. Article (CrossRef Link)
[20] Shojafar, M., Pooranian, Z., Naranjo, P. G. V., Baccarelli, E., "FLAPS: bandwidth and delay-efficient distributed data searching in Fog-supported P2P content delivery networks," *The Journal of Supercomputing*, pp. 1-22, 2017. Article (CrossRef Link)
[21] Cordeschi, N., Shojafar, M., Amendola, D., Baccarelli, E., "Energy-saving QoS resource management of virtualized networked data centers for Big Data Stream Computing," *Emerging Research in Cloud Distributed Computing Systems*, pp. 34, 2015. Article (CrossRef Link)
[22] Mukhopadhyay, D., Kulkarni, S., "An Approach to Design an IoT Service for Business Domain Specific Web Search," in *Proc. of Proceedings of the International Conference on Data Engineering and Communication Technology). Springer Singapore*, pp. 621-628, 2017. Article (CrossRef Link)
[23] Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., Tziritas, N., "A survey on text mining in social networks," *The Knowledge Engineering Review*, vol. 30, no.2, pp. 157-170, 2015. Article (CrossRef Link)
[24] Maynard, D., Li, Y., Peters, W., "Nlp techniques for term extraction and ontology population," in *Proc. of Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, IOS Press*, pp. 107-127, 2008. Article (CrossRef Link)
[25] Carenini, G., Cheung, J. C. K., Pauls, A., "MULTI-DOCUMENT SUMMARIZATION OF EVALUATIVE TEXT," *Computational Intelligence*, vol. 29, no. 4, pp. 545-576, 2013. Article(CrossRef Link)
[26] Bird, S., "NLTK: the natural language toolkit," in *Proc. of Proceedings of the COLING/ACL on Interactive presentation sessions Association for Computational Linguistics*, pp. 69-72, 2006. Article (CrossRef Link)
[27] Porter, M., "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980. Article (CrossRef Link)
[28] Taylor, A., Marcus, M., Santorini, B., "The Penn treebank: an overview," *Treebanks Springer Netherlands*, pp. 5-22, 2003 Article (CrossRef Link)
[29] Blei, D. M., Ng, A. Y., Jordan, M. I., "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003. Article (CrossRef Link)

Charu Virmani is an Associate Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Manav Rachna International University, Faridabad, Haryana, India. She is currently pursuing Ph.D. from YMCA University of Science and Technology, Faridabad, Haryana, India. Her subjects of interest include Computer Networks, Semantic Web and Social Networks.

Dr. Dimple Juneja is an Assistant Professor in the Department of Computer Applications, National Institute of Technology, Kurukshetra. She received Post-Doctorate from Louisiana State University, USA and Ph.D. in Computer Engineering from Maharishi Dayanand University, Rohtak. Dr. Juneja is Dale-Carnegie Certified Trained Teacher and is a recipient of Best Paper Award. Her subjects of interest include Multi-Agent Systems, Sensor Networks, Semantic Web and Distributed Operating System.

Dr. Anuradha Pillai is an Assistant Professor in the Department of Computer Engineering, YMCA University of Science and Technology, Faridabad, Haryana, India. She received Ph.D. in Computer Engineering from Maharishi Dayanand University, Rohtak. Her subjects of interest include Semantic Web, Web Mining and Social Networks.