DYNAMIC AND TEMPORAL USER PROFILING FOR PERSONALIZED RECOMMENDERS USING HETEROGENEOUS DATA SOURCES

Gokul S Krishnan⋆ *Sowmya Kamath S*†

⋆ gsk.it16f03@nitk.edu.in † sowmyakamath@nitk.edu.in
Department of Information Technology
National Institute of Technology Karnataka, Surathkal, Mangalore 575025 INDIA

ABSTRACT

In modern Web applications, the process of user-profiling provides a way to capture user-specific information, which then serves as a source for designing personalized user experiences. Currently, such information about a particular user is available from multiple online sources/services, like social media applications, professional/social networking sites, location based service providers or even from simple Webpages. The nature of this data being truly heterogeneous, high in volume and also highly dynamic over time, the problem of collecting these data artifacts from disparate sources, to enable complete user-profiling can be challenging. In this paper, we present an approach to dynamically build a structured user profile, that emphasizes the temporal nature to capture dynamic user behavior. The user profile is compiled from multiple, heterogeneous data sources which capture dynamic user actions over time, to capture changing preferences accurately. Natural language processing techniques, machine learning and concepts of the semantic Web were used for capturing relevant user data and implement the proposed "3D User Profile". Our technique also supports the representation of the generated user profiles as structured data so that other personalized recommendation systems and Semantic Web/Linked Open Data applications can consume them for providing intelligent, personalized services.

Index Terms**—** Dynamic user profile, Temporal profiling, Personalization, Recommender Systems, Semantic Web.

1. INTRODUCTION

The dramatic growth of information on the WWW has inadvertently led to information overload and hence finding a specific piece of information has become difficult and time consuming. As techno-savvy users transition from the the Social Web (Web 2.0) to the era of Semantic Web and Internet of Things (IoT), intelligent recommendation systems and user-specific services are indispensable. The popularity of these applications is partly based on the premise of extensive personalization to enhance user experience. The concept of personalization refers to the process of customizing applications and services, for tailor-made user experiences. This underscores the requirement for effective user profiling mechanisms that can capture both coarse-grained and fine-grained user preferences over time. Such user profiles may also be static or dynamic, where, the static profile never or rarely changes while dynamic profiles frequently change over time. Normally, users who actively consume application services can be modeled effectively by dynamic profiling techniques.

Some existing approaches [1] [2] [3] [4] [5] are of limited use due to the fact that their user profile generation methodology is dependent on a single data source or is geared towards a particular domain of service. For example, e-commerce is one such domain where personalization is pervasive. A user's profile and browsing/purchase history helps such websites to recommend more relevant products in an attempt to boost sales and revenue [6][7][8]. In e-learning and u-learning (ubiquitous-learning) environments, the focus is on understanding user's background, skill level, proficiency etc., so that their learning success can be enhanced [9][10]. Such specialized applications of user-profiling are highly focused and hence are not adaptable for other applications. Also, they often lack the capability to capture the temporal and dynamic nature of user activities. A significant drawback of these systems is that they fail to model user profiles that can capture their multiple persona, which may exist across different types of Web applications and services (like social, professional, hobby-related, political views and so on). When user profiles are modeled and created out of multiple user specific data sources, they are more complete and hence, more useful for personalization based applications. An added advantage is that the same profile can be used across applications to provide 'n' number of services that are tailored to user's needs.

In order that user profiles be available for consumption across applications, the primary requirement is to make any such data available in a standard and open format. The Semantic Web defines several frameworks for annotation, publication and consumption of open data based on metadata standards and Linked Open Data (LOD). LOD provides a method of publishing structured data so that it can be interlinked and are available for distributed querying by semantic applications. The Resource Description Framework (RDF) is the Web standard which is used to express resource level metadata in machine understandable manner. Another popular format that has extensive support of the Search Engine industry is JSON-LD (JavaScript Object Notation-Linked Data). JSON-LD enables encoding of Linked Data using JSON, thus allowing developers to use and serialize data in a simple, fast and efficient manner.

In this paper, we present an approach for generating enhanced user profiles that are compiled from multiple data sources, which are highly heterogeneous in nature. We introduce the concept of temporal trend analysis in the proposed approach for capturing the dynamic nature of user profiles. The generated user profiles are structured, dynamic and temporal in nature and can be published to the LOD Cloud to promote data reuse and interoperability, after taking appropriate measures to ensure that only user-approved applications with authentication tokens can consume the data.

The generated user profile can then be used for a number of applications and services like, question answering systems, search engines, knowledge representation and reasoning applications, and in general, artificial intelligence and Web Personalization applications. The remainder of this paper is organized as follows: Section 2 presents a discussion on the related work in the field of interest, i.e., user profiling for personalization. In section 3, we discuss the proposed approach and its implementation specifics in detail. The experiments and analysis of the effectiveness of the approach are presented in section 4, followed by conclusion and references.

2. RELATED WORK

As the volume and variety of Web data continues to increase exponentially, the task of finding relevant information fast, is increasingly difficult, adversely affecting user experience. This can hinder search experiences on e-commerce websites, Web search, e-learning etc. Recommendation systems have become increasingly ubiquitous in light of these problems. User profiling is the central premise of recommender systems, and much of the work in this area is this context.

Tao et al [1] proposed a personalized ontology model for users for the purpose of knowledge representation and reasoning over user profiles. The ability of users to read over a document and decide whether it is of interest to him or her is simulated with the help of ontologies referred to as *ontological user profiles* or *personalized ontologies* [1]. Tao et al's work also proposes that user background knowledge can be found in a better manner if both globally and locally analyzed knowledge can be integrated. The global knowledge or the world knowledge can also be said as the common sense knowledge which includes all the basic information and facts. The local knowledge includes user specific information which is private to the user. The personalized ontology created provides promising results. The advantage of this system is that the knowledge analysis is not restricted to any particular local or global domain, but includes both. The disadvantage is that the work assumes that all the local user repositories will have reference to the subjects which are present in the global knowledge base.

Skillen et al [2] proposed a novel approach for personalization of Help-On-Demand services in pervasive environments using Ontological user modeling and semantic rule-based reasoning. The personalization is achieved through the implementation of Ontological user profile modeling. Characteristics of a user are broken into lower granularity and modeled into the user profile data structure. The system successfully provides a model which can be applied onto Context-aware applications [2]. The limitation of this approach is that it overlooks the specific nature of user concepts in the ontological user profiling. Currently, the user profile deals with only those few features specified in the a generic profile. Zhao et al [3] proposed a method for integrating ontologies in Linked Open Data. The problems in Linked Open Data, mainly the heterogeneity of ontologies, are solved by the proposed framework system called Framework for InTegrating Ontologies (FITO) [3]. FITO uses the concepts of Graph Theory to integrate similar ontologies so that it is easy to fetch all the datasets within this ontology for semantic web developers. However, this system fails when a new ontology relationship comes into play in the Linked data, the change will not be reflected automatically in the integrated ontology as ontology matching is used for integration.

Hawalah and Fasli [4] proposed some methods to maintain the dynamic nature of user profiles for Web personalization. They tracked user interests by classifying it into short term and long term interests. The user's browsing data is collected from which interests are fetched. The authors propose 2 algorithms namely: Gradual Extra Weight (GEW) Algorithm and Contextual Concept Clustering (CCC) Algorithm. GEW algorithm decides how much weightage has to be given while adding terms in categories as user interests. CCC algorithm ensures that the user interests fall in the right category or context after the application of GEW algorithm. The advantage of the system is that the dynamic nature of user profiles is maintained to a very good measure. The drawback of this work is that ontological user profiles are not included. Existing ontologies are used to model user profiles, which may be lacking in effectively modeling the multitude of users.

Phuoc et al [5] put forward a novel approach towards compiling a live knowledge graph of connected things termed as Graph of Things (GoT). It uses other data sources from Linked Open Data as well along with data from the 'Things'. The SSN Ontology is used for the categorization of IoT sensors data and creates meaningful relationship mapping from IoT data to the interlinked LOD datasets [5]. However, the lack of user profiles and context awareness within the system doesn't make it user specific or intelligent. Another important drawback is the lack of security of the compiled data. There are no access restrictions specified, which means that anyone can access sensitive physical things data or user data. Grcar et al [11] designed a system that models user profiles based on his or her browsing history based on semantic Web data by grouping the pages visited by user into topics using k-means clustering. The user can view topics and pages associated with each topic. The most recently visited pages are clustered into one single group called the current interest. The advantage here is that interests of the user is grouped well, but the disadvantage is that the user profile is completely based on the interests and no other information is available in the user profile. Moreover, no other data source or activities of the user are considered.

Based on our observations, there are several avenues that can be explored for further improving user profiling techniques for recommenders and other personalization based applications. Various limitations of other works like, lack of attention to capturing the dynamic and temporal nature of users, across various heterogeneous data sources, usage of non-standard/limited ontologies and no support for data reuse are significant points that can affect the user profiling process. Our proposed approach tried to mitigate these problems by integrating existing ontologies to model the profile. The user profile generated using the proposed system can act as a very good knowledge base for personalization and recommendation systems or applications as it can be published and consumed via a variety of formats including a light-weight, open data format like JSON-LD to the LOD cloud.

3. PROPOSED APPROACH

The proposed user profile generation methodology is depicted in Figure 1. We describe each process in detail next. As the objective is to generate a unified user profile, obtaining the user permission is a prerequisite. This process is similar to the strategy adopted by most websites, when they intimate the user about the usage of cookies to track their website access, or when the session is tracked for a logged-in user. If the user has enabled single sign-on or OAuth [12][13], then the user data gathering process can be simplified further.

In the process pipeline, the first action by users is to authorize our tool to access their various account data, to the extent that the individual data-sharing APIs (of social networks like FaceBook, Twitter, LinkedIn etc) give access to. Another snippet of required data is the respective user's browsing history or session. The data obtained from these heterogeneous data sources is then processed further to compile the user's complete profile, which we refer to as the "3D Profile", which basically refers to the fact that the generated user profile incorporates all aspects of the user behavior, like browsing activities, social aspects, professional network, opinions and in-

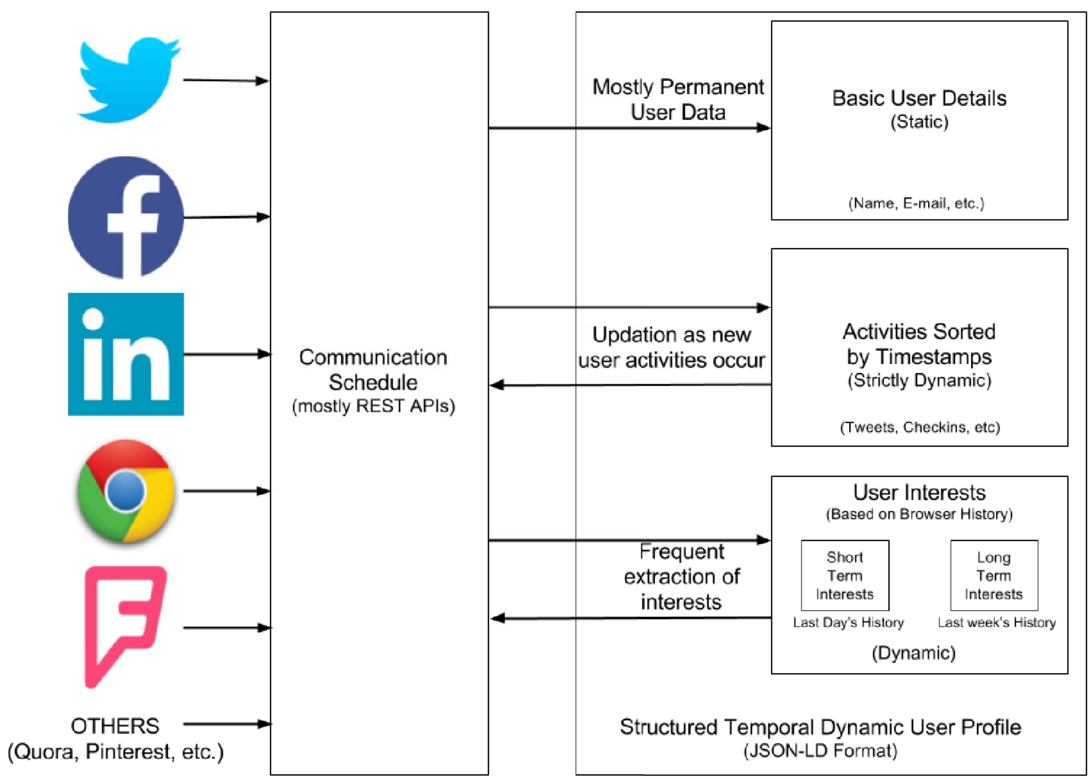

Fig. 1: Working of the overall system

terests. After the user provides permission to access his data from various sources, a 3D user profiling process consisting of static profile generation, The required user data from the various sources are integrated and a structured dynamic user profile is composed out of the integrated data sources. The user profile dynamically updates with new updated data which is streamed from the user linked data sources based on a temporal aspect. The created user profile, then acts as the knowledge base for personalization and recommendation systems. Each of these phases is explained in detail in sections 3.1, 3.2 and 3.3.

3.1. User Linking Process

The user linking process is performed with reference to all sources from which data is collected. Most such data sources provide REST APIs which can be invoked to extract a user's data, though restricted by the permissions given. Hence, a necessary condition is that the user has to grant full access to his data the first time he or she uses the system. Python API libraries and wrappers were used in scripts to extract the data from each source. In this implementation, Twitter and Foursquare were considered and the libraries used were Tweepy [14], a python library for twitter APIs, and Foursquare [15], a python wrapper for the foursquare APIs. Another source for understanding a user's preference is their Web browser history, which captures the browsing patterns and interests of the user. This history is extracted through a Chrome Extension (currently, only Chrome History is considered). The API chrome.history [16] was used in the extension to procure the history in JSON format, which will be added to the system's server at regular intervals. Each of these are stored in JSON format and the user's relevant data is extracted using a Python script, for initiating the process of compiling the user profile.

3.2. 3D User Profiling Process

Each user profile has 3 components - static, strictly dynamic and user interests (as depicted in Figure 1). The first component consists of all the data which mostly remains unchanged, like name, email id, gender etc, hence, is considered static in nature. The second component consists of all activities of the user sorted on the basis of timestamp at which the activity was performed. This component contributes to the dynamic and temporal nature of the user profile. The third and final component captures the user's interests and is updated frequently on the basis of browser history. This component is also dynamic and temporal in nature. These three components contribute significantly towards of the completeness of a user's profile, hence the generated output is called 3D profile.

For generating the static component of the 3D profile, details like name, gender, email, birthday, user-names used in various social media platforms etc are compiled into the user profile. These information come from different sources, name and email was taken from Foursquare data, birthday and Twitter username from Twitter data and so on. A Python script was used to extract the required data from the different data sources (JSON files), compile them into user-specific data dictionary called the 3D user profile dictionary. The FOAF (Friend Of A Friend) ontology [17] was used for the purpose of standardizing the user profile data. For example, to describe the metadata 'name', the relationship `FOAF.name` was used as the relationship between the user and the name value. Similarly, `FOAF.birthday`, `FOAF.mbox` and `FOAF.gender` properties were used for representing birthday, e-mail ID and gender in the 3D user profile.

In the second component, all the activities by the user were considered in a temporal manner. Each activity and the corresponding timestamps were extracted from the data sources using a Python script, and each activity was compiled as a dictionary and integrated into the same 3D user profile dictionary, generated earlier. This activity dictionary consists of a key 'activity_at' and its value as the timestamp. Additional knowledge extracted in this phase includes information on what a particular activity corresponds to. For example, if the user activity corresponds to tweet event, then the details of the tweet like tweet_id, its content, i.e., tweet_text are extracted and indexed. In case of Foursquare checkins, information like venue, geographical coordinates, venue type etc are extracted and indexed.

During the third phase, the process of inferring user interests is dealt with. The interests are discovered from the user's browser history by inspecting the webpages visited by the user over time, as user interests may change over time. For URL scraping, BeautifulSoup library was used [18]. The content of the webpage at each URL visited by a particular user is taken as a source for discovering the interests. To this, natural language processing techniques like tokenization, stopword removal and stemming are applied. For this purpose, NLTK library was used [19]. On this normalized content, the remaining terms are considered to be features which are fed into a machine learning classification algorithm to map to certain pre-defined common categories. The categories are then appended into the interests list present in the 3D user profile dictionary.

3.3. Interests Extraction Process

The process of extraction of interests is performed as part of the third component of the 3D user profile. The normalized scraped word features from the browsed URLs are fed into a machine learning classification algorithm which will categorize them into general categories. The machine learning classification algorithm used is a supervised learning algorithm which was trained with data consisting of words and their corresponding categories. A classifier called Generic Topic Classifier, available on a platform called monkeylearn [20] was used. The classifier used 2480 training samples and achieved an accuracy of 87%. Finally, for all new incoming word features scraped from the URLs, each was classified into these categories and the same categories were appended into the interests list present in the 3D user profile dictionary.

From the browser history, normalized content from the previous day's browsed URLs were taken and the categories after classification were considered as *short-term interests* and appended into the short-term interests list in the 3D user profile dictionary. For identifying a user's extended interests, the concept of *long term interests* was used. For this, the previous week's browser data was processed in a similar way and appended into the long-term interests list in the 3D user profile dictionary. The `FOAF.interest` relationship was used to describe the user interests metadata in a standard manner in the 3D user profile.

After the three components have generated their output, the integrated 3D user profile dictionary content is written into a JSON-LD file. A library called rdflib [21] was used to represent the generated user profile in JSON-LD format. This process basically serializes the base content into JSON-LD. This JSON-LD version can be published as linked data, and can act as knowledge base for supporting intelligent applications that intend to enhance user experiences using personalization or item recommendation.

3.4. Consuming the 3D User profile

The generated 3D user profile can be used for supporting a variety of application like Intelligent Information Retrieval (IIR), Question-answering (QA) and automatic information extraction (IE), to name a few. To demonstrate the effectiveness of our approach, a simple IE based search functionality was implemented, where one can enter a query and perform a search over his or her own data, after it is processed using the proposed approach. As soon as the user links his accounts, the system performs the various user profile generation processes and generates the user profile in JSON-LD format. The simple search functionality assists users in searching a specific concept in the newly generated structured data (3D user profile). To verify the extent of usefulness of the proposed approach, several experiments were performed, which are discussed in Section 4.

4. EXPERIMENTAL RESULTS

To analyze and justify the use of JSON-LD as the chosen format for representing the 3D user profiles, several experiments were designed to compare search performance over various data formats. The search was performed directly over a user's raw data, on the 3D user profile in JSON format, on the 3D user profile in RDF format and finally, on the 3D user profile in JSON-LD format. For performing search over the raw user data, JSON and JSON-LD formats, python scripts using JSON library functions were used. For RDF based user profiles, a python script using SPARQL queries based on graph concepts and the rdflib library functions was used.

Logically, when the search function is applied directly over raw user data, the time taken to match specific concepts will be large, as a match has to be determined after inspecting each data item separately with respect to user query. For example, consider a User 'A', who has 'm' data sources (files). Let us assume that the time taken for opening a file, creation of file pointers and other overhead for each file be 'x'. Let 'y' be the time taken for processing the combined content of all files. Assuming the worst case condition, the result to the user's search query lies in the 'm'th file. In this case, the time taken to generate results, denoted as T, if 'n' queries need to be performed to find the required data, will be:

$$T = n * m * x + y \tag{1}$$

In case of a preprocessed and compiled knowledge base, this overhead is eliminated. All files need to be processed only once during the compilation of the user profile, as they are stored as a dictionary. So, for the same search query result, the time T', is given by equation (2). Hence, empirically, a considerable amount of time savings can be achieved as all files need not be checked once the preprocessing and compilation of user profile is done.

$$T' = m * x + y \tag{2}$$

The actual observed results of the various experiments conducted are tabulated in Table 1. The runtime was recorded for a set of 8 queries over raw data sources, JSON 3D user profile, RDF 3D user profile and JSON-LD 3D user profile. Queries were chosen carefully to ensure that they covered all the various concepts covered by the profiles. Queries chosen were: 'tweets' (retrieves all the tweets by the user), 'checkins' (retrieves all checkins made by the user in Foursquare), 'name' (retrieves the full name of the user), 'email' (retrieves the email id of the user), 'birthday' (retrieves the birthday of the user), 'interest' (retrieves the interests of the user), 'sound' (searching for a particular term that is present in the user's tweet is given as query, so that a search is done of the full text of the set of tweets) and 'india' (retrieves all the checkins made by the user in India in his or her Foursquare). The raw dataset consisted of JSON files obtained directly from the APIs, without applying any of the processes designed as part of the proposed approach. These files were given as input to the proposed system to generate the temporal dynamic user profile in the JSON, RDF and JSON-LD formats. The experiments were performed on a machine with Intel Core i5 Processor, 4GB RAM, 128GB SSD.

From Table 1, it is evident that JSON-LD outperforms RDF and also that searching over RDF is a comparatively slower overall, in most cases. This is because of its graph structure and XML base of RDF. Due to this, we decided to represent the generated user profiles as JSON-LD rather than RDF. Also, it is to be noted that even though simple JSON also offers better result by a very low margin over JSON-LD, JSON-LD is chosen as it consists of metadata, that are beneficial to semantic applications. So, with a negligible cost in terms of a small margin in speedup, one can get structured, temporal, dynamic user profiles that also consists of metadata that can be consumed by semantics-based applications. As it can be seen from the tabulated results, the query 'interest' did not generate any output in case of query over raw data, as this is one aspect which is captured only by the 3D user profile.

Table 1: Query runtime over various formats (in Seconds)

Query	Raw	JSON	RDF	JSON-LD
'name'	0.630	0.00064	0.85543	0.00080
'email'	0.857	0.00064	0.83572	0.00094
'birthday'	0.113	0.00066	0.81255	0.00085
'tweets'	0.907	0.00068	0.78745	0.00076
'checkins'	0.412	0.00051	0.73621	0.00074
'interest'	-	0.64	0.75547	0.00086
'sound'	0.893	0.00062	0.77557	0.00086
'india'	0.914	0.00061	0.77465	0.00092

Table 2 illustrates the effective savings in time achieved when querying over the 3D user profile in JSON-LD format when compared to that over raw data . It can be seen that JSON-LD is almost 100% more faster than raw data implementation of the dynamic user profile. Table 3 shows the Speed Up of JSON-LD over RDF. Here again, it can be seen that there is significant savings in time in all cases.

Table 2: Speedup in time for JSON-LD over raw data

Query	Raw (time)	JSON-LD (time)	Speedup
'name'	0.63	0.0008	99.90 %
'email'	0.857	0.00094	99.88 %
'birthday'	0.113	0.00085	99.89 %
'tweets'	0.907	0.00076	99.90 %
'checkins'	0.412	0.00074	99.89 %
'interest'	-	0.00086	-
'sound'	0.893	0.00086	99.88 %
'india'	0.914	0.00092	99.88 %

Table 3: Speedup in time for JSON-LD over RDF

Query	RDF (time)	JSON-LD (time)	Speedup
'name'	0.85543	0.0008	87.30 %
'email'	0.83572	0.00094	89.01 %
'birthday'	0.81255	0.00085	94.77 %
'tweets'	0.78745	0.00076	91.62 %
'checkins'	0.73621	0.00074	82.03 %
'interest'	0.75547	0.00086	99.88%
'sound'	0.77557	0.00086	90.36 %
'india'	0.77465	0.00092	89.93 %

Figure 2 shows a comparative plot of observed time to query over the raw data and the JSON-LD user profile format. It can be seen that, once the user profile has been generated for a user, every operation made on it is highly optimized, when compared to those performed over raw data. Hence, it can be concluded that the proposed approach was effective, not only in terms of improving processing time, but also in providing reusable data in a structured format, that is both temporal and dynamic in nature. The JSON-LD format used provides an added bonus in the form of associated metadata, which makes building semantic applications that consume the 3D user profile, easier.

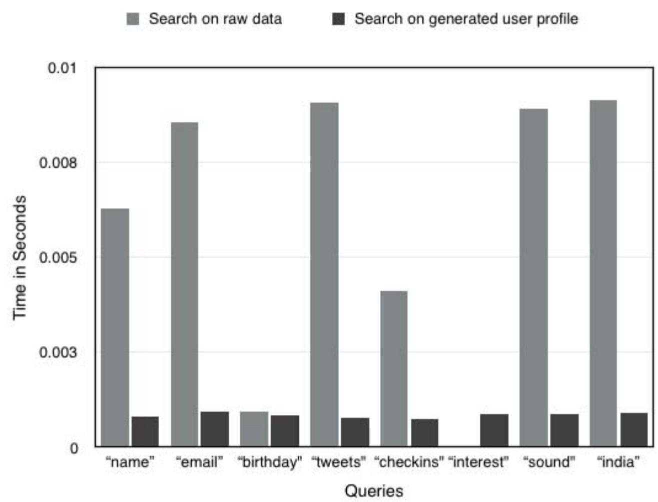

Fig. 2: Time Taken for querying raw data vs querying over the generated 3D user profile

5. CONCLUSION & FUTURE SCOPE

In this paper, a novel method for generating structural, dynamic and temporal user profiles from heterogeneous data sources was presented. The generated user profile can act as a knowledge base for personalization and recommendation systems as they can be published as linked data. For validating the efficacy of the proposed approach, a simple search application was applied to the various user profile formats generated, including raw data. Experimental results showed that the proposed 3D user profile represented in JSON-LD format was the better, compared to others, and effectively captured an additional dimension of user behavior that is ignored in most approaches, that is, temporal user interests. The main future directions for the proposed approach is in the area of security and privacy. As user's private data is handled, any concerns have to be specifically addressed by incorporating adequate encryption techniques to ensure security and privacy of user data. Machine Learning or Deep Learning for extraction of user interests can be an area of possibilities. A full fledged application, which is rich in semantics, personalization and recommendation features, will be developed and tested so as to harness the power of structural, dynamic and temporal user profiles in realizing true personalization in Web applications.

6. REFERENCES

[1] X. Tao, Y. Li, and N. Zhong, "A personalized ontology model for web information gathering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 496–511, April 2011.

[2] Kerry-Louise Skillen, Liming Chen, Chris D. Nugent, Mark P. Donnelly, William Burns, and Ivar Solheim, "Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments," *Future Generation Computer Systems*, vol. 34, pp. 97 – 109, 2014, Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence.

[3] Lihua Zhao and Ryutaro Ichise, "Ontology integration for linked data," *Journal on Data Semantics*, vol. 3, no. 4, pp. 237–254, 2014.

[4] Ahmad Hawalah and Maria Fasli, "Dynamic user profiles for web personalisation," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2547 – 2569, 2015.

[5] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Hung Ngo Quoc, Tuan Tran Nhat, and Manfred Hauswirth, "The graph of things: A step towards the live knowledge graph of connected things," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3738, pp. 25 – 35, 2016.

[6] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12 – 32, 2015.

[7] Konstantinos Christidis and Gregoris Mentzas, "A topic-based recommender system for electronic marketplace platforms," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4370 – 4379, 2013.

[8] Jian Wang and Yi Zhang, "Opportunity model for e-commerce recommendation: Right product; right time," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2013, SIGIR '13, pp. 303–312, ACM.

[9] Aleksandra Klanja-Milievi, Boban Vesin, Mirjana Ivanovi, and Zoran Budimac, "E-learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education*, vol. 56, no. 3, pp. 885 – 899, 2011.

[10] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper, *Recommender Systems in Technology Enhanced Learning*, pp. 387–415, Springer US, Boston, MA, 2011.

[11] Dunja Mladeni Miha Grar and Marko Grobelnik, "User profiling for interest-focused browsing history," in *Proceedings of the SKIDD 2005*. 2005, SKIDD 2005, ACM.

[12] "Web authorization protocol (oauth)," `https://datatracker.ietf.org/wg/oauth/charter/`, Accessed: 2017-04-01.

[13] "The oauth 2.0 authorization framework," `https://tools.ietf.org/html/rfc6749`, Accessed: 2017-04-01.

[14] "Tweepy," `http://www.tweepy.org/`, Accessed: 2017-04-01.

[15] "Foursquare api client for python," `https://github.com/mLewisLogic/foursquare`, Accessed: 2017-04-01.

[16] "chrome.history - google chrome," `https://developer.chrome.com/extensions/history`, Accessed: 2017-04-01.

[17] "Foaf vocabulary specification," `http://xmlns.com/foaf/spec/`, Accessed: 2017-04-01.

[18] "Beautiful soup," `https://www.crummy.com/software/BeautifulSoup/`, Accessed: 2017-04-01.

[19] "Natural language toolkit," `http://www.nltk.org/`, Accessed: 2017-04-01.

[20] "General topic classifier," `https://app.monkeylearn.com/main/classifiers/cl_5icAVzKR/tab/live-tab/`, Accessed: 2017-04-01.

[21] "Rdflib," `https://rdflib.readthedocs.io/en/stable/`, Accessed: 2017-04-01.