Using Hypergraph-based User Profile in a Recommendation System

Hilal Tarakci and Nihan Kesim Cicekli
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

Keywords: User Modeling, User Profile, Hypergraph-based User Model, Knowledge Representation, Recommendation System.

Abstract: We propose a hypergraph-based user profile which facilitates aggregating partial profiles of the individual and obtain a complete, multi-domain user model. The aggregation involves a semantic enhancement procedure which results in enriched user profiles. The proposed user model is capable of extracting general and domain-based user profiles and answering several connected data queries such as recommendation, in reasonable time. In this paper, we present a recommendation case study which uses the proposed user model and illustrate the traversal algorithms for a variety of connected data problems.

1 INTRODUCTION

The popularity of social networking sites has dramatically increased over the last decade. The user's profile can be extracted by examining the individual's behavior (Gauch et al., 2007). The user's activities on social websites reveal important information about his/her profile. Social networks differ in nature and are used for various purposes. For instance Facebook is used for social interaction and entertainment whereas LinkedIn is used only for professional interests. Therefore, mining separate social networks independently results in partial profiles of the user which merely represent user's interests for one or few domains. Seamless aggregation of partial user profiles obtained from different knowledge sources is still an unsolved problem. In this paper, we present the implementation of a hypergraph-based user model to aggregate partial profiles of the individual to obtain a complete, semantically enriched, multi-domain user model and show that it can be used for different recommendation purposes.

The employed user profile structure is mutually associated with the aggregation methodology. The aggregation process depends on the predefined user model data structure, and this structure is defined according to the main goals of the aggregation. If the main purpose is producing an interoperable user model, the profile is generally defined by a standard (Orlandi et al., 2012) or user-defined (Wischenbart et al., 2012; Ghosh and Dekhil, 2008) ontology. In this paper, one of our main goals is solving connected data problems such as recommendation effortlessly. An effective solution strategy for connected data problems is matching an entrance point to the data structure and traversing the neighbours according to the specified algorithm. Therefore, graphs naturally support connected data problems (Robinson et al., 2013). The vertices usually represent the items and the users where an edge between a user and an item indicate user's interest on that item. The edges could be associated with weights which represent the strength of the relation between the vertices. Since the graph is only capable of representing binary relations, other approaches have been proposed for handling higher order relations in user modelling domain. There are a few studies which define user model as bipartite (Tiroshi et al., 2013) and tripartite graphs (Chen et al., 2012). In general, if the number of vertex types n is known in advance and the relations in the user model are binary, an n-partite graph is capable of representing the profile. However, if there are higher-order relations, a hypergraph is more appropriate to represent the user model (Li and Li, 2013; Kramar et al., 2013; Bu et al., 2010).

In a previous paper we presented the initial ideas for using hypergraph in the modelling of user profiles (Tarakci and Cicekli, 2012a; Tarakci and Cicekli, 2012b). In this paper, the main contributions are *(i)* a user profile representation with hypergraphs, *(ii)* a semantic aggregation methodology and *(iii)* a recommendation case study to illustrate the solution for various connected data problems.

The paper is organized as follows. Section 2 summarizes the related studies. Section 3 defines the proposed hypergraph based user model and presents the

Tarakci H. and Kesim Cicekli N..
Using Hypergraph-based User Profile in a Recommendation System.
DOI: 10.5220/0005029600270035
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 27-35
ISBN: 978-989-758-049-9

recommendation case study. The evaluation details are presented in Section 4. Section 5 concludes the paper by summarizing the future work.

2 RELATED WORK

The aggregation of partial user profiles includes several issues such as entity matching, resolution of duplicates and conflicts, and heterogeneity of the partial user profiles (Orlandi et al., 2012). Furthermore, the objective of the aggregated user model influences the aggregation strategy. In literature, there are diverse aggregation approaches.

In (Abel et al., 2013), form-based and tag-based profiles are managed separately. The former is a list of attribute-value pairs whereas the latter is a set of weighted tags. The aggregation strategy for form-based profiles is unifying sets of attribute-value pairs. Heterogeneous attribute vocabularies is resolved by using an alignment function which maps profiles to unified attribute-value space. However, this alignment function may result in duplicate entries in the final user profile. Moreover, when there are conflicts in the aggregated profiles, both values are included in the result. The aggregation of tag-based profiles is accomplished by taking a weighted accumulation of partial tag-based profiles. The authors do not consider aggregating tag-based profiles and form-based profiles with each other. In our paper, we do not make such a distinction. We seamlessly aggregate received partial user profiles by taking their weighted accumulation. We solve heterogeneous vocabulary problem by using Freebase [1].

In (Orlandi et al., 2012), the aim is to obtain an interoperable, source-independent, multi-domain user profile. Therefore, the aggregated user pofile is represented by using popular standard ontologies. During aggregation the authors address the problem of recurring items and calculating a global weight for them. To achieve this, they keep track of *provenance data* which is the metadata for the user profile item such as the source of the item and the timestamps. Keeping track of origins of interest relations enables the recalculation of item weights during aggregation of the partial profiles. We also keep track of the provenance data by storing the knowledge source, the short term profile date and the exact keyword of the item. We extend this information each time the item and the user is bound together.

In (Wischenbart et al., 2012), the aggregation is handled by semi-automatically extracting schema from social web data and integrating the extracted schemata with existing integration tools. In (Plumbaum et al., 2011), an aggregation ontology is proposed to semi-automatically aggregate partial user profiles.

In this paper, the objective of the aggregation is two-fold: *(i)* to obtain a user model based on a hypergraph which reduces connected data problems such as recommendation into graph traversal algorithms and *(ii)* increasing recommendation accuracy with the proposed semantic enhancements.

In literature, the semantic enrichment is accomplished by disambiguating the concept by linking to an external vocabulary, using a secondary vocabulary when the concept could not be linked, enriching the concept by adding sysnsets, expanding the concept by retrieving related concepts from the external vocabulary according to a predefined treversal algorithm, by using friends or like-minded users' profiles as explained in the survey (Abdel-Hafez and Xu, 2013). We achive semantic enhancement by using a middle ontology in front of the external vocabulary and calibrating the middle ontology concepts according to system requirements.

Most user modelling and recommendation problems are connected data problems. Connected data problems are solved by generating appropriate traversal algorithms which traverse the sub-graph related to the problem. It is claimed that graph databases are faster than relational and NoSQL databases when dealing with connected data (Robinson et al., 2013) since relational and NoSQL databases lack relationships causing connected data problems to be costly on these databases. In graph databases, a traversal query performance depends on the size of the sub-graph which is going to be traversed. That is, the size of the whole graph does not effect the traversal performance. Therefore we use a graph database for the implementation of the hypergraph in this paper.

In (Tiroshi, 2012; Cena et al., 2013), graph based user models are presented. In (Tiroshi, 2012), concepts are linked to each other by examining an external ontology; therefore the nodes could be traversed in a generic way. In (Cena et al., 2013), horizontal propogation amongst siblings and vertical propogation amongst ancestors and decendants are defined. Our approach is an enhanced version of the former approach. We connect semantically related concepts to each other during aggregation process and our predefined node labels and edge types enable defining more specific traversal algorithms easily.

[1] Freebase, https://www.freebase.com/

3 DATA MODEL AND AGGREGATION

3.1 Preliminaries

A hypergraph is the generalization of an ordinary graph by introducing hyperedges which are non-empty subsets of the vertex set (Gallo et al., 1993). Vertices of a hypergraph represents the entities to be modelled such as people and concepts. Hyperedges represent the high order relations between those entities.

Besides hypergraphs, there are *property graphs* which contain key-value property pairs (Robinson et al., 2013). In a property graph each node and edge can have multiple key-value pairs whereas in a hypergraph, an edge can connect more than two nodes. Every hypergraph can be represented by a property graph by adding extra key-value pairs to annotate nodes which are connected by the same hyperedge. Thus, property graphs are identical to hypergraphs in terms of representation power. In this paper, we use property graphs in the implementation, since the graph database we adopted[2] supports property graphs.

In the property graph, properties can be indexed by using a tree like structure. Therefore, a two step search on graph can be adopted: First the concept is located in the index structure and then with this *short-cut* to the graph, traversal algorithm can be applied. In graphs, the cost of local read operations is constant, since adjacent vertices and edges are already connected.

3.2 Hypergraph User Model

We collect short term profiles for registered users from predefined knowledge sources such as facebook, linkedin for predetermined time periods. Besides, we allow users to add their interests manually via an interface. In this paper, we focus on constructing a holistic, multi-domain user model by aggregating the received short term profiles by utilizing the proposed hypergraph data structure. We use the term *partial profile* and *short term profile* interchangeably in the paper.

The main components of the user model is summarized in Table 1. In the proposed framework, *users*, *items* and *domains* are represented with distinct node types U, I and D. The supported *domains* are predefined. Freebase commons package [3] is used as domains. A domain starter node $D_{[d]}$ is created for each Freebase domain.

In the proposed model, different types of relations are represented by different edge types. E_{bind} is the edge with label *InterestedIn* and connects a user u to an item i to represent that *"u is interested in i"*. In order to model the semantic relations between items, E_{inner} is used and the label of the edge represents the nature of the semantic relation. For instance, in Figure 3 *ContributedTo* edge is an E_{inner} edge which indicates the start node contributes to the end node of the relation. The item i is connected to its belonging domain d by using E_{domain} edge.In the proposed model, items without any domains are not allowed, every item must be connected to at least one domain starter node. The friendship between users is represented with E_{friend} edges. E_{inner} and E_{domain} edges enable content-based recommendations where E_{friend} supports collaborative recommendations.

Table 1: Our Hypergraph User Model.

Notation	Description	Type
u	a user	Node
U	Set of users	Hyperedge
i	an item	A Node
I	Set of items	Hyperedge
$D_{[d]}$	Domain starter node for each domain d	Node
E_{bind}	Metadata for user-item (interest) relation	Hyperedge
E_{inner}	The semantic relation between items	Hyperedge
E_{domain}	The domain bind between domain starter node and items	Hyperedge
E_{friend}	Friendship between users	Hyperedge
H_u	General (long term) user profile	A sub hypergraph

Definition. Hypergraph User Profile. The hypergraph user profile H_u is the aggregated, semantically enhanced user model for the user u (Eqn.1). It is the union of the user's friends whom the user follows or is followed by (Eqn. 2), the user's explicit profile which is the set of user's declared interested items (Eqn. 3) and the user's semantically enhanced profile (Eqn. 4) The user's enhanced profile is defined as the set of items whose shortest path to the user node has at least *min*, at most *max* steps.

[2] Neo4j, http://www.neo4j.org/

[3] Freebase, https://www.freebase.com/

$$H_u(u;min;max) = U_{friends}(u) \cup U_{explicit profile}(u) \cup U_{enhanced profile}(u;min,max) \quad (1)$$

$$U_{friends}(u) = u \xrightarrow{follows} (u_f) \cup (u_f) \xrightarrow{follows} u \quad (2)$$

$$U_{explicit\ profile}(u) = u \xrightarrow{interestedIn} (i) \xrightarrow{isInDomain} (d) \quad (3)$$

$$U_{enhanced\ profile}(u;min;max) = u \xrightarrow{*min..max} (i) \xrightarrow{isInDomain} (d) \quad (4)$$

Basically the hypergraph user model consists of sets of nodes and strongly typed hyperedges. The proposed hypergraph consists of nodes for domains, interest items and users; and edges for explicitly stated interests, semantic relationships between interest items and domain relations of the items. As an example scenario, assume that there are three users whose names are *GraceKelly*, *IngridBergman* and *TippiHedren*. *IngridBergman* states interest in three items: *Alfred Hitchcock* who is a director and *Alfred Hitchcock Presents* and *The Twilight Zone* which were popular TV shows in 1950s. *GraceKelly* expresses interest in the director *Alfred Hitchcock* whereas *TippiHedren* does not declare any interest. Also these three users are friends. The hypergraph which models the illustration scenario is in Figure 1; for clarity friendships and domains are eliminated. The implementation of this hypergraph actually corresponds to the property graph shown in Figure 2.

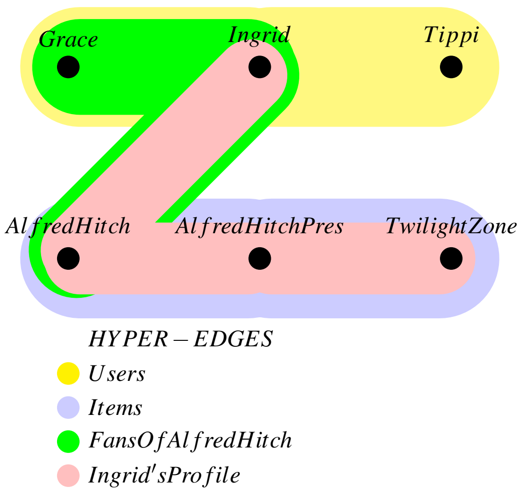

Figure 1: Illustration Scenario in Hypergraph.

In the hypergraph (Figure 1), the yellow hyperedge models the set of users, whereas in the property graph (Figure 2) the users are represented with red nodes. Similarly, the blue hyperedge in the hypergraph is a wrapper for the set of items where the blue nodes in the property graph are item nodes. The pink hyperedge in the hypergraph links *Ingrid* with her declared interested items. In the property graph, this hyperedge is modeled by connecting *Ingrid* to the items with an edge type *InterestedIn*. All users are connected to each other via *following* mechanism to represent their friendship. The type of the edge between users is *Follows* and the type of edge between a user and an explicitly declared item is *InterestedIn*.

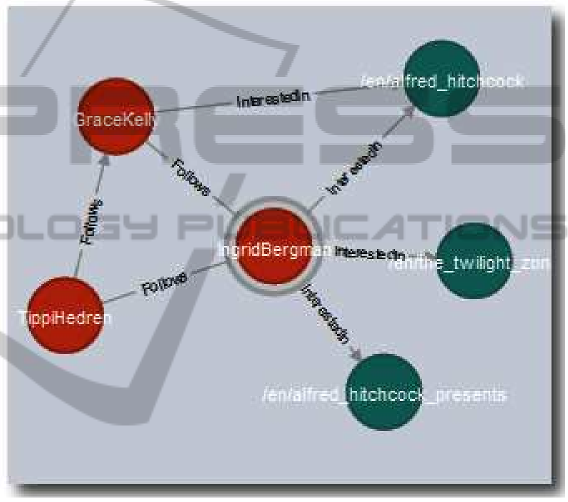

Figure 2: Illustration Scenario in Property Graph.

When a new keyword expressing the user's interest arrives for aggregation, the keyword is located in the external knowledge base. In this paper, we use Freebase as the knowledge base and a *disambiguation routine* which processes the keyword if the keyword does not match any entity in Freebase. The disambiguation routine performs several text pocessing operations. For example it replaces the special characters with the nearest letters in English alphabet such as replacing *ş, ç* by *s, c*; removes the terms such as *"Fans Of"*, *"Quotes"* from the keyword; splits the keyword if it contains characters such as "&, /". Freebase search api returns matching concepts ordered by score, therefore we used the first concept with the highest score as the matching entity for the keyword.

We defined a *domainizer routine* to assign the disambiguated concept to the domains it belongs. In the proposed model, Freebase domains which corresponds to Freebase commons package is used. For each domain type, a starter domain node is created at system initiation. The type information of the concept is retrieved from Freebase. The retrieved type information not only includes domain knowledge, but also more specific type information. For instance, when the type informtion of *Alfred Hitchcock* is retrieved, types such as *Film director, Film producer, Film writer* are also retrieved under the type *Film* which is a domain. We exploit those specific types to compute the weight of the domain. In other words, we build a weighted domain structure by accumulating specific types under each domain. Afterwards, we prone the weighted domain structure according to the predefined *domain threshold* and relate the concept with the most frequent domains by using an edge with type *IsInDomain*. In Figure 3, the purple nodes represent the domain starter nodes. There is one starter node for each domain and all of the items belonging to that domain is related to that node. This design facilitates domain-based queries.

The *semantic enhancement* of a concept is achieved by retrieving predefined *Freebase Metaschema* properties which provides higher order relations between concepts. Metaschema ontology consists of 46 properties and constructs another layer over huge Freebase ontology which has over 3500 properties. Metaschema connects important information and eliminates excessively detailed semantics in Freebase. We further reduced 46 properties to 9 properties by considering their benefits in user modelling and apply a threshold on the number of retreived relations. The 9 properties we support for semantic enhancement include *BroaderThan/ NarrowerThan, ContributedTo/ HasContributor, Created/ CreatedBy, HasGenre/ GenreOf, HasName/ NameOf, HasChild/ HasParent, PractitionerOf/ HasPractitioner, HasSubject/ SubjectOf, SuperclassOf/ SubclassOf*. Using Freebase over a middle ontology enables writing domain-independent or domain-configured algorithms by using different thresholds for different domains. For instance, *ContributedTo* and *Created* properties reveal important information for *Film* and *Music* domains where *ChildOf* property is meaningful in *People* domain. The concepts retrieved during semantic enhancement are related to the key concept with an edge of type named after the metaschema property linking them. For instance, in Figure 3, *Alfred Hitchcock* which is represented by the blue node at the center is related to his movies, TV shows and songs with an edge of type *ContributedTo*.

3.3 Recommendation Case Study

Various connection-based queries could be answered by defining traversals on the proposed hypergraph data structure.

Traversal Example 1. In order to obtain the *user domain-based model* for the user u and domain d, the user is located in the external index system for users and the user node in the hypergraph is reached with a *short-cut*. Eqn. 5 computes user domain-based model by matching the items which are in domain d and have a shortest path with the user u with length at most *max*.

$$P_{domain}(u;d;max) = u \xrightarrow{*0..max} (i) \xrightarrow{IsInDomain} d \quad (5)$$

The json output for the query *"Retrieve the domain based profile for user GraceKelly for domain TV."* contains the user's declared interest *Alfred Hithcock* and the items in her enhanced profile such as the TV show *Alfred Hitchcock Presents* and its several episodes.


```
{ "data": [
    { "row": [
        "GraceKelly",
        "Alfred Hitchcock"
      ] },
    { "row": [
        "GraceKelly",
        "Alfred Hitchcock Presents"
      ] },
    { "row": [
        "GraceKelly",
        "The Case of Mr. Pelham"
      ] },
    ...
  ] }
```


To obtain the *general user profile*, during *Traversal Example 1* domain is not included as a parameter to the traversal function.

Traversal Example 2. In order to discover the users interested in a domain d, the set of users that have shortest path with length at most *max* to d are retrieved (Eqn. 6).

$$U_{domain}(d;max) = d \leftarrow (i) \xleftarrow{*0..max} (u) \quad (6)$$

Traversal Example 3. To discover users interested in an item i, the set of users that have shortest path with length at most *max* to i are retrieved (Eqn. 7).

$$U_{item}(i;max) = i \xleftarrow{*0..max} (u) \quad (7)$$

Traversal Example 4. The ability to discover related concepts of an item i in other domains as in Eqn. 8 enables answering questions such as *"What are the films about Nasa?"* or *"Find biographies about Mozart."*.

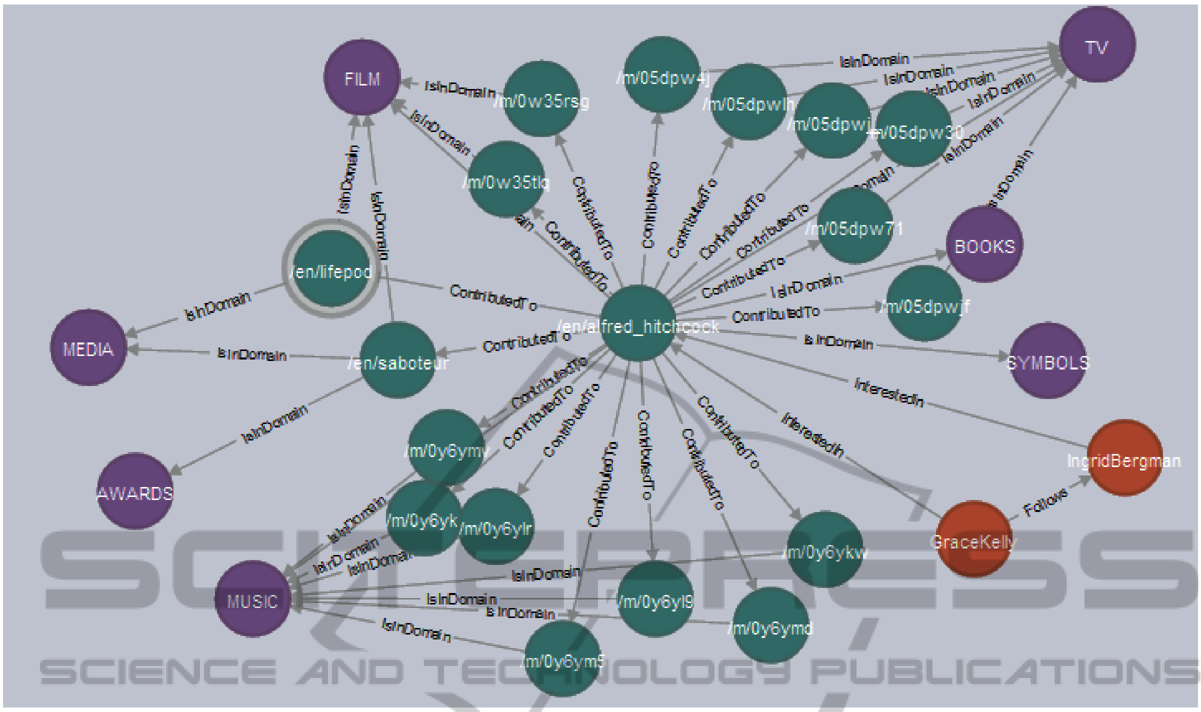

Figure 3: A Sample User Model.

$$
\begin{aligned}
R_i(i;max) = {} & i \xrightarrow{IsInDomain} (d_1) \\
& and \;\; i \xrightarrow{[*2..max]} (d_2) \\
& and \;\; (otherItem) \rightarrow d_2 \\
& and \;\; d_1 \neq d_2
\end{aligned} \tag{8}
$$

Traversal Example 5. In order to calculate a user's interest on a concept, shortest path algorithms could be applied as in Eqn. 9.

$$I_{interest}(u;i) = shortestPath(u,i) \tag{9}$$

Figure 4 shows the interface of the recommendation system that we implemented based on these traversal algorithms. In the illustration scenario (Figure 2), *GraceKelly* declared one interest item: director *Alfred Hitchcock*.

The interface is divided into six columns. The first column shows the friendship information, the second column enables manual addition of an interest item and shows the users declared interests. The number next to the declared interest is the frequency of that item and it is incremented by one whenever the same concept is matched with different keyword-information source pairs. The list next to the frequency information shows the domains of the item. The third column exposes the domain aggregation for the user.

The fourth and fifth columns show the top 15 recommendations for the user. *Random recommendations* part recommends any item which is connected to the user in the graph via other items or users. *Detailed recommendations* part recommends items that are connected to the user's declared items and ranks the recommendation by checking two factors: the number of declared items of the user which constitute a path of length 2 between the user and the recommended item and the accumulated frequency of the items in that path. For instance, there are two paths of length 2 between *IngridBergman* and *Mystery* item over the user's two declared interests: *The Twilight Zone* and *Alfred Hitchcock Presents*. Since both items are assigned frequency 1, the accumulated frequency is 2.

Popular recommendations part recommends items only in popular domains and eliminates other domains. Path length ordering is applied. *Far recommendations* part recommends items at least three, at most five steps away from the user. The sixth column computes whether the user is interested in the specified item and lists the users who might be interested in. For instance, in Figure 4, *GraceKelly*'s interest for *Marnie*, which is a movie directed by Alfred Hitchcock, is over declared interest *Alfred Hitchcock* and the path length is 2.

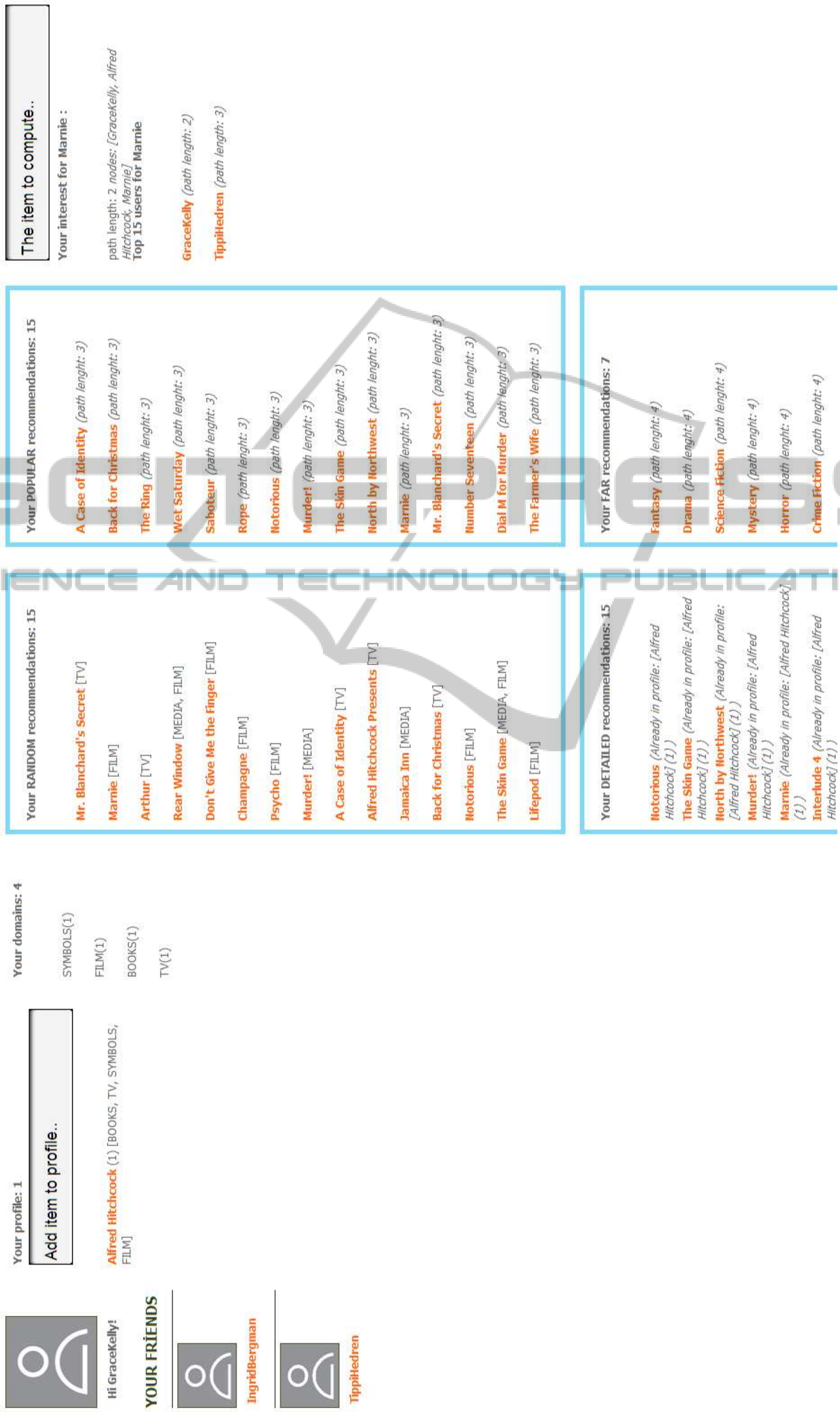

Figure 4: Fun Guide Interface.

4 EVALUATION

The dataset is prepared by collecting social web activities of 204 users on their Facebook accounts. The page likes are the main information source. Besides keywords from shared videos, checkins are also collected. In the populated graph database, there are a total of 22746 nodes of which 204 are user nodes and 22466 are item nodes. Most popular domains are *music, awards, people, business, media* and *film* and the most unpopular domains are *library, skiing, zoos and aquariums, bicycles* and *physics*.

The dataset is split into training and testing sets. 80% of each partial profile is located in the training set and the remaining 20% is left for testing set. During evaluation, for each keyword on each partial profile, the keyword is disambiguated by using Freebase. Keywords which can not be disambiguated are skipped. After disambiguation, the domains are decided by the domainizer routine by taking the domain threshold as 4. In other words, each item is linked to at least 1, at most 4 domains. Then semantic enhancement procedure is applied by taking the threshold 5. Taking those thresholds higher increase the semantic nature of the constructed model, but the overall processing time for evaluation also increases. During evaluation, if the item in the test set is already connected before the interest is declared, this is considered as a hit. In the evaluation, we considered the ratio of number of hit items to the number of items of the user in the short term profile. For 204 users, the average of hits-to-total items ratio is calculated as *0.61*. In the baseline, the knowledge base usage and enhancement is removed and 40 users of the same dataset is evaluated likewise. The average hits-to-total ratio for the baseline is *0.25*. The resulting scores show that usage of a knowledge base and the enhancement procedure successfully predicts the user's future interests. The domain and semantic enhancement thresholds are kept small to obtain fast evaluation, increasing them would result in a better hits-to-total ratio in future interest prediction.

Moreover, 20 users of the same dataset is evaluated for cold-start. During evaluation for cold-start, each user is extracted from the dataset and the hypergraph is populated with the remaining users. Afterwards, during aggregation of the user to the previously populated hypergraph, hits-to-total radio is calculated. For 20 users, the average of hits-to-total items ratio is calculated as *0.52*. In the baseline, the average hits-to-total ratio is *0.03*. The resulting scores show that usage of a knowledge base and the enhancement procedure successfully predicts the user's future interests in cold-start as well.

In addition, the recommendation study is under human evaluation currently. The user is able to connect with his/her Facebook account and LinkedIn account. The system provides *Import Facebook* and *Import LinkedIn* functionality and aggregates the obtained partial profiles. The user rates the recommendations provided by the system. The human evaluation system is going to be online for 1 month.

5 CONCLUSIONS

In this paper, we presented a framework for aggregating partial user profiles into a holistic, multi-domain user model. The main objective of the aggregation is to obtain a user model data structure which reduces the connected data problems such as recommendation into defining graph traversal algorithms.

Graphs naturally support connected data problems and using property graphs which are equivalent to hypergraphs makes definition of graph traversal algorithms easier by providing filtering mechanisms such as node labels and edge types. In other words, it is possible to write traversal algorithms specific to a label or an edge type without traversing irrelevant nodes or edges in the hypergraph. Another goal of our system is to successfully predict user's future interests. To achieve this goal, we used an external knowledge base via a middle ontology and configured the use of middle ontology according to user modelling domain. We only used properties in the middle ontology such as *ContributesTo, Creates, SuperclassOf etc.* that are relevant to user modelling domain.

During evaluation, we showed that the system could predict future interests of the user with a hit-to-total ratio of *0.61*. If the semantic enhancement and usage of external knowledge base is eliminated, the score would be *0.25*. As future work, we are going to categorize users according to social web usage habits, separate long term and short term user profiles and evaluate the framework against a bigger dataset.

ACKNOWLEDGEMENTS

This work is partially supported by The Scientific and Technical Council of Turkey Grant "TUBITAK EEEAG-112E111".

REFERENCES

Abdel-Hafez, A. and Xu, Y. (2013). A survey of user modelling in social media websites. *Computer and Infor-*

mation Science, 6(4):59–71.

Abel, F., Herder, E., Houben, G.-J., Henze, N., and Krause, D. (2013). Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209.

Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L., and He, X. (2010). Music recommendation by unified hypergraph: Combining social media information and music content. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 391–400, New York, NY, USA. ACM.

Cena, F., Likavec, S., and Osborne, F. (2013). Anisotropic propagation of user interests in ontology-based user models. *Inf. Sci.*, 250:40–60.

Chen, B., Wang, J., Huang, Q., and Mei, T. (2012). Personalized video recommendation through tripartite graph propagation. In Babaguchi, N., Aizawa, K., Smith, J. R., Satoh, S., Plagemann, T., Hua, X.-S., and Yan, R., editors, *ACM Multimedia*, pages 1133–1136. ACM.

Gallo, G., Longo, G., and Pallottino, S. (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2):177–201.

Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). The adaptive web. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, chapter User Profiles for Personalized Information Access, pages 54–89. Springer-Verlag, Berlin, Heidelberg.

Ghosh, R. and Dekhil, M. (2008). Mashups for semantic user profiles. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1229–1230, New York, NY, USA. ACM.

Kramar, T., Barla, M., and Bielikova, M. (2013). Personalizing search using socially enhanced interest model built from the stream of user's activity. *J. Web Eng.*, 12(1-2):65–92.

Li, L. and Li, T. (2013). News recommendation via hypergraph learning: encapsulation of user behavior and news content. In Leonardi, S., Panconesi, A., Ferragina, P., and Gionis, A., editors, *WSDM*, pages 305–314. ACM.

Orlandi, F., Breslin, J. G., and Passant, A. (2012). Aggregated, interoperable and multi-domain user profiles for the social web. In Presutti, V. and Pinto, H. S., editors, *I-SEMANTICS*, pages 41–48. ACM.

Plumbaum, T., Schulz, K., Kurze, M., and Albayrak, S. (2011). My personal user interface: A semantic user-centric approach to manage and share user information. In Smith, M. J. and Salvendy, G., editors, *HCI (11)*, volume 6771 of *Lecture Notes in Computer Science*, pages 585–593. Springer.

Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph Databases*. O'Reilly, Beijing.

Tarakci, H. and Cicekli, N. K. (2012a). Ubiquitous fuzzy user modeling for multi-application environments by mining socially enhanced online traces. In Masthoff, J., Mobasher, B., Desmarais, M. C., and Nkambou, R., editors, *UMAP*, volume 7379 of *Lecture Notes in Computer Science*, pages 387–390. Springer.

Tarakci, H. and Cicekli, N. K. (2012b). Ucasfum: A ubiquitous context-aware semantic fuzzy user modeling system. In Filipe, J. and Dietz, J. L. G., editors, *KEOD*, pages 278–283. SciTePress.

Tiroshi, A. (2012). Graph based user modeling. In Duarte, C., Carrio, L., Jorge, J. A., Oviatt, S. L., and Gonalves, D., editors, *IUI*, pages 371–374. ACM.

Tiroshi, A., Berkovsky, S., Kaafar, M. A., Chen, T., and Kuflik, T. (2013). Cross social networks interests predictions based ongraph features. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 319–322, New York, NY, USA. ACM.

Wischenbart, M., Mitsch, S., Kapsammer, E., Kusel, A., Prll, B., Retschitzegger, W., Schwinger, W., Schnbck, J., Wimmer, M., and Lechner, S. (2012). User profile integration made easy: model-driven extraction and transformation of social network schemas. In Mille, A., Gandon, F. L., Misselis, J., Rabinovich, M., and Staab, S., editors, *WWW (Companion Volume)*, pages 939–948. ACM.