Augmented Context-Based Conceptual User Modeling for Personalized Recommendation System in Online Social Networks

Ammar Alnahhas, Faculty of Information Technology Engineering, Damascus University, Syria
Bassel Alkhatib, Faculty of Information Technology Engineering, Damascus University, Syria

ABSTRACT

As the data on the online social networks is getting larger, it is important to build personalized recommendation systems that recommend suitable content to users, there has been much research in this field that uses conceptual representations of text to match user models with best content. This article presents a novel method to build a user model that depends on conceptual representation of text by using ConceptNet concepts that exceed the named entities to include the common-sense meaning of words and phrases. The model includes the contextual information of concepts as well, the authors also show a novel method to exploit the semantic relations of the knowledge base to extend user models, the experiment shows that the proposed model and associated recommendation algorithms outperform all previous methods as a detailed comparison shows in this article.

KEYWORDS
Knowledge Graph, Recommendation, Social Networks, User Modeling

INTRODUCTION

Online social networks are getting more attention nowadays, they grew rapidly in the last few years, and attracted many users, as they can communicate, share and get more information by using these networks.

As more people are engaged to the online social networks, huge amount of data is posted every day, millions of tweets on Twitter and posts on Facebook are generated causing an overwhelming stream of data. Social networks are widely used to share news, updates and events, but as the data is getting larger; users are more likely to miss important items that may be interesting to them, this issue rise the importance of building personalization solutions that can guide users to interesting content of social web including posts, people or pages.

In this paper we present a personalized recommendation system that helps users of social networks find the best textual content that matches their interest using semantic analysis of the text with help of a knowledge base. To identify the user interest, a conceptual user modelling methodology that exploits the contextual relations of concepts is introduced. As well as a recommendation algorithm
that matches the user model with the content representation. There has been a series of work in this area, to our knowledge all of previous works that introduced conceptual models of social web users use named-entities to represent the interest of the users, whereas in our work we propose to represent the text as a series of concepts that contains the representation of words, phrases and entities in text, we present conceptual user model where users are mapped to concepts, and contextual conceptual user model where users are represented as a graph of contextually connected concepts. We also try to exploit the semantic relations of the knowledge graph to expand the user model, so we propose a learning-based approach that finds the importance of each semantic relation type in the field of recommendation. Our results compared with the state-of-the-art methods show that the proposed model effectively out-perform all other models.

This paper is structured as follows, Section 2 contains a detailed view of related work, Section 3 presents the problem definition and contribution, text representation method is explained in Section 4, in Section 5 we show the experiment setup, Section 6 shows the conceptual user model and Section 7 shows the contextual conceptual user model, Section 8 elaborates the model expansion approach, Section 9 presents the comparison of our models with previous models, and Section 10 concludes the paper.

RELATED WORK

Building recommendation systems for online social networks has attracted many researchers in the last few years, a few researchers aimed at studying content recommendation, but many user modelling techniques are used in other researches aiming at link, news or ‘who to follow’ recommendations. We can classify the researches in three different categories:

1. **Collaborative filtering**: Where items are recommended to a user by considering users with similar interests, or by considering similar items to items already user interested in;
2. **Statistical content based**: Where users are modelled according to the textual content of their items, the content is processed statistically like in IR systems, and items are recommended if its content is similar to the user model;
3. **Semantic analysis of content**: Where users are modelled according to the semantics extracted from textual contents of their interest, item are recommended if its content is semantically similar to the user model.

Collaborative Filtering Methods

The main approach of collaborative filtering is to use the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users (Su & Khoshgoftaar, 2009), so when two users share many interested items, we can predict interest in new items for one of them if the other is interested in them, item based collaborative filtering depends on the fact that if two items share the same user interests then new users who are interested in one of them is more likely interested in the other.

Many researches targeted the field of collaborative filtering to build recommendation systems for social networks, some researchers (Chen et al., 2012) propose to use collaborative ranking technique to find correlation between users and items so that if users post similar content to social network; similar items can be recommended for them in the future, while other researchers rely on social associations (Vosecky, Leung, & Ng, 2014) to build collaborative user model, or use ego networks (Sun & Zhu, 2013) to link users so that homophily between them can represent the collaboration. Similarly, a graph-theoretic model (Yan, Lapata, & Li, 2012) is proposed that ranks items and their authors according to the relations between them. Some researches use tags as the main source of collaboration, either by linking tags to users and items (Guy, Zwerdling, Ronen, Carmel, & Uziel,
2010), or by building a tag map (Xiao, Du, Zhu, & Li, 2012) that describes the relation between tags; where tags are connected if they are used by a single user or by friends, or by using social tag prediction (Yuan, Huang, Sun, Li, & Xu, 2015) and interest evolution model of tags that find scores of tags in items. A signal-based model (Arru, Feltoni Gurini, Gasparetti, Micarelli, & Sansonetti, 2013) to represent users is proposed, which includes a time dimension in the representation of the user interests, users similarity is calculated using signal processing techniques and user-to-user collaboration model is applied for recommendation.

Collaborative filtering methods suffers from many drawbacks, including the need to analyze huge and dynamic networks that are very complex to be done in real time. Moreover, cold start problem is well known in these methods, especially in the case of social networks as the huge amount of data makes it impossible to get collaborative information, because as soon as sufficient information is available for a piece of data, it will be outdated and new data will arrive instead.

**Statistical Content-Based Methods**

In these methods textual content of social network is processed to generate a statistical model for each user, text is tokenized into words; and methods like bag-of-words are used, then a statistical model such as TF-IDF pairs (Pennacchiotti, Silvestri, Vahabi, & Venturini, 2012), Hashtag frequency (Abel, Gao, Houben, & Tao, 2011a; Ma, Jia, Xie, & Lin, 2015; Zhou, Wu, Chen, Chen, & Ying, 2014), LDA (Khater & Elmongui, 2015; Kim & Shim, 2014) [6, 11, 12] or sentimental features (Cui, Du, Shen, Zhou, & Li, 2017) is applied. In these methods, a model for each user is built according to the content he is already interested in, then a matching algorithm is used to estimate the interestingness of the user in a new item.

While a simple IR method is used (Benzarti & Faiz, 2015) to build a personalized recommendation system where user models are built from user related content in various social networks using TF-IDF, other research (Pennacchiotti et al., 2012) extend this approach to use pairs of terms as a unit for modelling users, along with terms from user friends, similar approach is proposed in (Chen, Nairn, Nelson, Bernstein, & Chi, 2010) where authors build a URL recommendation system that compare content of URL pages to user models built form joining user content with content of user followers. Authors in (Elmongui et al., 2015) combine the user content with his social activities, they use a classifier to find a topic for each item and then use machine learning approach to match new items with users. A different orientation in the literature is to use tags and especially hashtags to model user interests, authors (Abel et al., 2011a) build a temporal user model as a set of weighted tags where weights are found by their frequency and timestamp, while in (Ma et al., 2015) a multi-tag correlation approach is proposed, tags are linked to users and to each other, then user-tag correlation is updated by using tag relations, similarly, Researchers in (Zhou et al., 2014) build tag-user graphs for computing the similarities between microblogs and users. Another approach (Kim & Shim, 2014) is to use probabilistic modeling based on generalized LDA model as well as matrix factorization. Authors of (Cui et al., 2017) used the sentimental information to help with recommendation, they incorporate it into the traditional content-based method.

Modeling user by bag-of-words methodology will not catch the semantics of the text, therefore recommender systems will only consider vocabulary that has already been part of the user model, so, content with similar meanings will not be recommended as relations deduced from knowledge bases are not utilized in these methods.

**Semantic User Modelling Methods**

Many researchers build recommendation systems for social networks that uses named-entities to represent user model (Abel, Hauff, Houben, & Tao, 2012; de Graaff, van de Venis, van Keulen, & Rolf, 2015; Kapanipathi, Jain, Venkataramani, & Sheth, 2014; Karidi, Stavarakas, & Vassiliou, 2016; Lu, Lam, & Zhang, 2012; Orlandi, Breslin, & Passant, 2012; Piao, 2016; Piao & Breslin, 2016a, 2016b, 2016c; Zarrinkalam, Fani, Bagheri, Kahani, & Du, 2015), they try to represent the semantics
of the text by extracting entities that are usually DBPedia (Auer et al., 2007) concepts and use them to represent the content, some of researches (Piao & Breslin, 2016b) leverage the semantic relations to extend the user model and make recommendations more suitable, while others (Abel, Gao, Houben, & Tao, 2011b) try to build a semantic user model based on twitter posts by linking them to mainstream news articles to enrich the tweet text, they introduced semantic user model as entity-based one, the entities are extracted from user own tweets and the related news articles, whereas in (de Graaff et al., 2015) they introduced a point of interest recommendation system based on geotagging. Places and attractions are recommended to users depending on their semantic model built from content they post on the social web. Authors of (Orlandi et al., 2012) focuses on building an Interoperable user profile that can be used across many social networks, they use DBPedia entities to represent users, they compared two types of entities to use: resources and categories. Authors of (Piao & Breslin, 2016b) and (Piao, 2016) propose a link recommendation system by using named entities as well, they suggested enriching user models by three types of connected concepts from DBPedia: categories, classes and entities, they investigated the effect of temporal dynamics of user interests as well. While in another research (Piao & Breslin, 2016a) they investigated the role of aggregating user models form different social networks on the accuracy of the recommendation, they also investigated user model extension using categories and classes of DBPedia. Authors of (Lu et al., 2012) also used DBPedia entities to represent user interests but they suggested to enrich user models by random walks in the knowledge graph. In (Zarrinkalam et al., 2015) semantic topics method is proposed where temporally correlated entities are grouped into topics that are not represented yet in the knowledge base. Researchers in (Kapanipathi et al., 2014) propose to represent user interest as a hierarchical interest graph, they show an algorithm to build user profile based on their interests being leaves of DBPedia hierarchy, they use spreading activation theory to expand the interest up. Authors of (Karidi et al., 2016) proposed to use a hierarchical knowledge graph to represent user interests, each user profile is represented as a tree where nodes of the tree are classes from a known taxonomy. Besides in (Piao & Breslin, 2016c) authors proposed to use Wordnet Synsets along with DBPedia entities to represent user interests, they show better results than entity model used in previous works, therefore, this is a clear indicator that named-entity representation of user interests is not sufficient and should be improved. Therefore, using extracted named entities to represent content of text is not enough to catch the semantics inside the text, besides using entities cannot represent the context of the text that can be an important factor in modelling users. In our work, on the contrary of all previous works, we suggest to represent the textual content of an item in a social network as a series of consequent concepts that represent not just entities, but also all other types of words and phrases of the text using common-sense meanings, besides we suggest to use the contextual relations between concepts to represent user interests which is—to our knowledge—a novel work in the field of social web recommendation.

**Problem Definition**

The goal of our work is to find the most suitable semantic modelling methodology of user interests in social networks in order to provide content recommendations that matches every user. Formally, given a user model \( u \) and a text \( t \):

**Definition 1:** User interest measurement function \( I(u,t) \): is a function that rates the text \( t \) according to user with model \( u \) such that the function reflects how much this text is interesting to the user and how much he likes to see it in the social network.

The recommendation system fetches items from user timeline and re-ranks them in a descending order according to function \( I \), so that content with high interestingness value for the user appears first. Our work aims at finding the best modeling strategy and best user interest measurement function exploiting semantics of textual content and the context of the text.
To be able to represent the full semantical meaning of the text and to maintain the context of the semantics, we should represent all parts of it as semantic blocks, where nouns, verbs, named entities that can be represented as concepts in a knowledge base should be considered. To be able to include as much content of the text as possible we should use a very rich background knowledge base, so we choose to use ConceptNet (Speer, Chin, & Havasi, 2017) which is a multilingual knowledge graph that connects words and phrases with labeled relations called assertions, the key advantage of ConceptNet is that it focuses on the common-sense meanings of words and not just named-entities, moreover, it is built from many sources such as Open-source common sense, WordNet, OpenCyc and DBPedia, that makes it a rich knowledge base that covers most terms and phrases.

The contribution of our work can be summarized as follows:

- We provide a new semantic representation of text in social networks as a series of concepts;
- We suggest a novel contextual semantic user modelling methodology and show its effectiveness in representing user interests and provide a recommendation algorithm over this model;
- We provide a method to exploit semantic relations to enrich user models that is suitable for recommendation applications.

Conceptual Representation of Text

As we described earlier, we are going to represent the text as a series of consequent concepts extracted from ConceptNet, ConceptNet concepts are associated with words or phrases, so it is easy to convert a part of the text to a concept when this part corresponds to the textual representation of this concept. We suggest a simple greedy algorithm to convert text into series of concept: first the text is tokenized, stop words are removed and hashtags are normalized to ordinary words, then the result list of words are processed with alg.1, the intuition behind the greedy algorithm is that phrases represents the semantic of its words so the longer the text part is the more accurate the semantic is, moreover, natural language structure tend to link words with its predecessors, so if adding a word to a meaningful phrase makes a new longer meaningful phrase so it is better to add it.

Algorithm 1: input: series of words – output: series of concepts

```plaintext
s is an empty string
C is an empty string
For each word w in input
    If s + w represent a prefix of some concept:
        s <- s + w
        if s represents a concept:
            c <- s
        else if c is not empty
            add c to the output
        make c and s empty
```

To be able to find if a string represents a prefix of some concept, we convert all available concept representations into a prefix tree where each representation is tokenized and words are used to denote edges of the tree.

As ConceptNet is a multilingual knowledge base, we consider only the English concepts, so as a preprocessing phase, we process the flat file of ConceptNet that contains all assertions and extract the English concepts, assertions that connect two English concepts are only regarded. Table 1 shows statistics about English part of ConceptNet.
We choose English concepts because it is the largest language represented in ConceptNet and it is easy to collect dataset for experiment user, however, it is rational that our method is cross-lingual and should work for any language but further testing and validation should be carried out as a future work.

**EXPERIMENT SETUP**

In this section, we are going to describe the dataset and the performance measures we are going to use to test the different algorithms we are going to present in the coming section of this paper.

Twitter is an online social network that is widely used around the globe, as Twitter provides an open and easy-to-use API that can be used to get users, relations and content we are going to test our approach on a dataset extracted from Twitter.

To collect the data, we choose 980 arbitrary users who have more than 2000 tweet posted before the date of collection, the users was chosen to post English tweets as we are using the English part of ConceptNet. Using twitter API, we get the last 3200 tweets for each user at most which is the limit the API imposes. For each tweet we normalize that text by removing links and converting hashtags into words, then tokenize the text and apply algo.1 on each tweet to get the concept list representation, we keep track of both the word representation and the concept representation for comparison with other methods, Table 2 shows statistics of the collected data.

**Table 1. Statistics about English content of ConceptNet**

<table>
<thead>
<tr>
<th>Number of concepts</th>
<th>1507812</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of assertions</td>
<td>3098816</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of users</th>
<th>980</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tweets</td>
<td>1771648</td>
</tr>
<tr>
<td>Average tweets per user</td>
<td>1808</td>
</tr>
<tr>
<td>Number of distinct words</td>
<td>1069125</td>
</tr>
<tr>
<td>Number of distinct concepts</td>
<td>181214</td>
</tr>
</tbody>
</table>

After investigating the dataset, we notice that users are not consistent in posting tweets, that is, some users post too many tweets per day whereas some other users post a tweet every week, so, as content has temporal association, we choose a subset of users that are consistent with average tweets per time unit as well as the total number of tweets they have, the compact dataset contains 224 users with statistics shown in Table 3, the remaining users data is used to extract knowledge base relation weights as will be described in Section 8.

The evaluation metrics used in our experiments are similar to these used commonly in recommendation systems and are derived from information retrieval system evaluation metrics, as the recommendation process is similar to the process of searching for content that is suitable for a user, besides, users will check the first k results of recommendation in a similar behavior of information retrieval behavior. We measure algorithms performance using the following metrics:

**MRR:** which is 1/x where x is the rank of the first interesting item recommended to the user.
Table 3. Compact dataset statistics

<table>
<thead>
<tr>
<th>Number of users</th>
<th>223</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tweets</td>
<td>474092</td>
</tr>
<tr>
<td>Average tweets per user</td>
<td>2125</td>
</tr>
<tr>
<td>Number of distinct words</td>
<td>381415</td>
</tr>
<tr>
<td>Number of distinct concepts</td>
<td>114863</td>
</tr>
</tbody>
</table>

**P@K**: precision at rank K, which is the ratio of interesting items in the first K recommended items.

**R@K**: recall at rank K, which is the number of interesting items in the first K recommended items divided by the total number of interesting items.

**F@K**: which is defined in terms of precision P and recall R as following:

\[
F = \frac{2 * P * R}{P + R}
\]  

The compact dataset is divided into two parts: the training part which constitute 90% of the dataset, it was created by taking first 90% of each user tweets ordered by date, the remaining 10% of the dataset are used for testing, the testing part is distributed equally among users, so a random recommender will result in almost an equal result for each user, P@10 of a random recommender which recommend items randomly will be about 0.04.

The ground truth is that each user is interested in the content he posts, so to evaluate a recommender; its result is compared to the testing data where the true items are the user own items and the false ones are the items of all other users. This can roughly reflect the interest of users because users may have not seen some content that they are also interested in, where we account it as not interesting because he did not post it, this problem cannot be solved in automatic testing scheme and needs a pilot user study to address it, however, if an algorithm gets good precision with ground truth data, it should get at least the same precision in real data because the actual true set of each user contains the true set we consider which is the items posted by the user, that means the precision we get is guaranteed to be the minimum true one when evaluated by real users.

**CONCEPTUAL USER MODEL**

To test the efficiency of modelling users using the concepts of ConceptNet, we define the user model as a vector of weighted concept, this model is used in many related works, where items of the model were either words or named entities, whereas we suggest to use concepts extracted in Section 2.

**Definition 2**: Conceptual user model $M_u$ of user u is defined as:

\[
M_u = \{(c, w_c) \text{ for each } c \in C\}
\]  

where $C$ is a set of all known concepts, $w_c$ is a weight that reflects the interestingness of the user u in the term, where:

\[
W_c = T_c * P_c
\]
is the number of items containing \( c \) that are posted by the user, and \( P_c \) is a popularity factor; where this factor is high when the concept is not popular, this is important for recommendation as it is more desirable to recommend more specific concepts rather than common ones that are popular. We suggest using inverse user frequency to represent the factor \( P_c \) as follows:

\[
P_c = \log(U/U_c)
\]

where \( U \) is the total number of users in the dataset and \( U_c \) is the number of users who are interested in concept \( c \). The user model is normalized, so that the sum of all weights equal to 1.

The recommendation algorithm assigns an interestingness value \( V_i \) for each item \( i \), then the items are ranked according to this value. First, we represent each Item \( i \) as a series of concepts \( CL_i \), the value \( V_i \) is a normalized sum of \( W_c \) for each \( c \) in \( CL_i \):

\[
V_i = \frac{1}{k} \sum_{c \in CL_i} W_c
\]

where \( k \) is the number of concepts in \( CL_i \).

Running the algorithm on the testing data shows effective results, the MRR is 0.693175, Figure 1 shows P@k, R@k, S@k and F@k diagrams for this model.

**Figure 1. Results for conceptual model recommendation system**

[Graphs showing P@k, R@k, S@k, and F1@k for different values of k]

**Contextual Conceptual User Model**

Representing interests of a user as a weighted list of concepts proved to be efficient, however, it is important to consider the context that each concept is related to in order to model the interests of a user more accurately. So, it is more suitable to link each concept with the other concepts that appear in its context in the user training set, the more a set of concepts appear in the context of each other
in the training set of a user, the more this user is interested in the set of these concepts as a whole. To address the context of each user we provide the following definition:

**Definition 3:** Contextual conceptual user model: for a user \( u \) we define a contextual conceptual user model as a weighted graph \( G_u \) where \( N \) are the nodes, and \( E \) are the edges, each node of \( G_u \) represents a concept from the knowledge base, if a concepts \( c1 \) and \( c2 \) appear in the same context in the user data then there is an edge that connects the nodes representing \( c1 \) and \( c2 \).

The weight \( W_e \) of an edge \( e \) connecting concepts \( c1 \) and \( c2 \) represents the interestingness of user \( u \) in items in which \( c1 \) appears in the context of \( c2 \), and can be calculated as the following:

\[
W_e = \frac{1}{k} \cdot T_{c1,c2} \cdot P_e
\]  

(6)

where \( k \) is the number of edges in the user graph, \( T_{c1,c2} \) is the number of items in the training dataset where \( c1 \) appears in the context of \( c2 \), and \( P_e \) is a popularity factor representing how common is this edge among users, we suggest to calculate \( P_e \) as following:

\[
P_e = \log \left( \frac{U}{U_e} \right)
\]  

(7)

where \( U \) is the total number of users in the dataset and \( U_e \) is the number of users that \( e \) is an edge in their graphs.

To represent an Item \( I \) of a social network, supposing \( I \) is represented as a sequence of concepts denoted by \( CL_i \), \( I \) is represented as a fully connected unweighted graph \( G_i \) where each concept in \( CL_i \) is represented by a node and there is an edge between each two nodes.

To find the interestingness value \( V_{u,I} \) of user \( u \) with an item \( I \) we sum the weights of the Intersection of \( G_u \) and \( G_i \), so that if an item context conform to user model then it is interested to this user, formally we can find \( V_{u,I} \) as follow:

\[
V_{u,I} = \frac{1}{k} \sum_{e \in G_u \cap G_i} W_e
\]  

(8)

where \( k \) is the number of edges in \( G_i \). Note that the intersection of \( G_u \) and \( G_i \) is a new graph \( G \) where node set of \( G \) is the result of intersection of node sets of \( G_u \) and \( G_i \) and edge set of \( G \) is the result of intersection of edge sets of \( G_u \) and \( G_i \) with weights adopted from \( G_u \) as \( G_i \) is unweighted.

The recommendation algorithm ranks each item \( I \) for user \( u \) according to \( V_{u,I} \), then the result is viewed to the user. Testing the algorithm using our dataset shows that it is very efficient, the MRR is 0.816321. Figure 2 shows details about the result of this user model and associated recommendation algorithm.

**Extending Conceptual Model**

Each concept in the knowledge graph is connected to other concepts using semantic relations, intuitively, if a user is interested in a concept he may be interested in related concepts. Taking this into consideration; it is worth investigating the association between semantic relations and user modelling in order to improve the result of recommendation systems.
Concepts in ConceptNet are connected using 47 different types of semantic relations such as “Is a”, “part of” and “synonym”. It is obvious that not all of these relations are suitable to be exploited to extend user interests model, so we introduce a method to find the importance of each relation type for the purpose of recommendation. The method uses machine learning approach to learn the importance of each relation, so we choose a part of the dataset that does not overlap with compact dataset explained in Section 3 as a training set. Formally we are calculating a distribution $D$ that can be represented as a vector where each relation type corresponds to a dimension, that is, $D = (d_r, d_{x_r}, \ldots, d_{y_r})$ where $d_r$ represent the relative importance of relation type $r$.

For each user we find the extended training set $TS_u$, first we find $SKG_u$ a subgraph of the knowledge graph that includes all concepts from user model extracted in Section 4, the intuition is that if a relation type is important for recommendation, it should appear frequently in $SKG_u$ of many users. The extended training set $TS_u$ of the user $u$ can be defined as following:

$$TS_u = \{c : \exists E(x, c) \in SKG_u\}$$ \hspace{1cm} (9)

$TS_u$ is a set of concepts where there is an edge in $SKG_u$ ending at it, intuitively, we assume that if $V_{c_l}$ is the interestingness of concept $c_l$ for a user and there is an edge $e(c_1, c_2)$ in $SKG_u$ where this edge relation type is $r$, then:

$$V_{c_2} = V_{c_1} \cdot d_r$$ \hspace{1cm} (10)

For a concept $c$ in $TS_u$ we have already calculated a weight $V_c$ for it as this concept is part of the user model (see Section 4) by definition, whereas we can find the training weight $nV_c$ for this concept by using a generalized form of Equation (10), because each concept in $TS_u$ may have multiple edges ending at it:
\[ nV_c^2 = \sum_{e \in \{c_1, c_2\} \in SKG_u} V_{c_1} \cdot d_{r(e)} \]  

(11)

For the value of \( D \) to be perfect, the value of each concept \( V_c \) should be close to the target value \( nV_c \), more accurately; if we represent the conceptual user model as a vector \( V \) and the training user model that consists of target values as a vector \( nV \) where number of dimensions equals to the total number of concepts, then these two vectors should be identical for perfect values of \( D \). We consider the cosine distance between these two vectors as a measure of distance. Hence, we would like to find the values of \( D \) such that the we maximize the following formula for each user:

\[ \frac{V_n V}{|V| \cdot |nV|} \]  

(12)

Hence, we would like to maximize the following formula taking into account all users:

\[ \frac{1}{|U|} \sum_{u \in U} \left( \frac{V_u nV_u}{|V_u| \cdot |nV_u|} \right) \]  

(13)

where \( U \) is the set of all users. The vector \( nV_u \) can be represented as a multiplication of a matrix \( M_u \) by the vector \( D \):

\[ nV_u = M_u \cdot D \]  

(14)

The matrix \( M_u \) has dimensions \( 47 \times C \) where \( C \) is the number of concepts in \( TS \) of the user, and 47 is the number of types of semantic relation, each item in \( M_u \) can be calculated as:

\[ M_{ij} = \sum_{e \in \{c_1, c_2\} \text{ where } r(e) = j} V_c \]  

(15)

where \( r(e) \) is the relation type of edge \( e \). So the formula to maximize is:

\[ \frac{1}{|U|} \sum_{u \in U} \left( \frac{V_u (M_u \cdot D)}{|V_u| \cdot |M_u \cdot D|} \right) \]  

(16)

To find the value of \( D \) that maximize the Formula 16, it can be addressed as an optimization problem that can be handled using a global search algorithm. We suggest to use a genetic algorithm to find the best value of \( D \), each individual of the population represents a different instance of vector \( D \) and the fitness function is the Formula 16, these are the steps of our algorithm:

Algorithm 2

1. Generate 1000 random vectors to constitute the population
2. Find the fitness of each individual by using Formula 16
3. Normalize each individual so that $\sum d_i = 1$

4. Repeat until no more significant improvement:
   a. Select 1000 individuals using tournament selection method
   b. Generate new generation using two-point crossover
   c. Do some arbitrary mutation for some individuals
   d. Evaluate new individuals using by using formula 16
   e. Normalize each individual so that $\sum d_i = 1$

5. Choose best individual to be the solution.

The algorithm converges to best solution that is shown in Figure 3. To exploit the semantic relations, we are going to show two experiments: augmentation of conceptual user model and augmentation of contextual conceptual user model.

Figure 3. The importance of each relation type as the result of Algorithm 2

To augment conceptual user model with new concepts that may be interesting for the user we define the extended conceptual user model as:

$$EM_u = \left\{(c, w_c) \text{where } \exists x \in M_u \text{ and } \exists e(x, c) \in G \right\} \cup M_u$$

(17)

where $G$ is the knowledge graph, and $w_c$ is:

$$w_c = w_x * d_{i(c)} * P$$

(18)

$r(e)$ is the relation type of edge $e$, and $P$ is a penalty factor that can be found according to popularity of concept $c$. 
Figure 4 shows the result of augmenting the conceptual user model, we can observe how the results are improved by this extension.

Augmenting the contextual conceptual user model involves adding edge to the user graph $G_u$ depending on edges in the knowledge graph. If two concepts $c1$ and $c2$ are connected in $G_u$ and there is an edge $e(c1,c3)$ in the knowledge graph then we can add the edge $e(c2,c3)$ to $G_u$ with weight derived from weight of $e(c1,c2)$ and the type of relation $e(c1,c3)$. Formally, Extended user graph $EG_u(N_{eg},E_{eg})$ can be formed as following:

$$E_{eg} = E_u \cup \{ (x,y,w_{x,y}) \mid \exists e(x,a) \in G_u \text{ and } \exists e(y,a) \in G \}$$

$$w_{x,y} = \text{Max}_{\text{for each } a \text{ where } e1(x,a) \text{ and } c2(a,y)} \left( w_{e1} \ast d_{c2} \ast P \right)$$  \hspace{1cm} (19)
The node set \( N_{re} \) is the union of original user model nodes with the set of new nodes added by new edges. \( P \) is a popularity factor and can be found using \( IUF \).

Figure 5 shows the results of augmented contextual conceptual user model compared to model without augmentation. Besides, Table 4 compares the MRR values for different models.

**Table 4. MRR for different user models**

<table>
<thead>
<tr>
<th></th>
<th>Conceptual Model</th>
<th>Extended Conceptual</th>
<th>Contextual Model</th>
<th>Extended Contextual</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRR</td>
<td>0.693175</td>
<td>0.730058</td>
<td>0.816321</td>
<td>0.823518</td>
</tr>
</tbody>
</table>

**Comparison With Other Methods**

In order to show the efficiency of our described models, we are going to compare the results with the most successful previous works along with some baselines and state of the art models.

All models are implemented and tested using the same dataset, the models are the follows:

- **Tf-idf model**: Baseline model that uses the IR measures to model each user, new items are ranked according to tf-idf;
- **Pairs frequency model (PFM)**: This model is presented in (Pennacchiotti et al., 2012), where the authors uses statistical model that improves tf-idf model by using pairs of words instead of words themselves;
- **LDA model**: As described in (Khater & Elmongui, 2015) and (Kim & Shim, 2014) where each user is represented as a distribution of latent topics, based on the topic distribution generated by training data;
- **Entity model (EM)**: Using this model users are represented as a vector of DBPedia entities and is used with variation in many recent researches (Orlandi et al., 2012; Piao, 2016). To generate DBPedia entities for our dataset, we use TAGME Web API (Ferragina & Scialetta, 2010) that can extract parts of the text that correspond to Wikipedia pages, that is a DBPedia named entities;
- **Steiner tree model (STM):** That is presented in (Karidi et al., 2016) where the researchers extract a Steiner tree for each user given the knowledge graph that is represented as a taxonomy and user nodes extracted from user items. To implement this model, we first convert ConceptNet into a simple graph by eliminating all relation types but the “IsA” relation, then a Steiner tree is built for each user to extend his model, the Steiner tree is traversed using reversed topological sort and weights are assigned to new nodes, then these nodes are added to original user model.

Table 5 shows the results of our models: conceptual model (CM), augmented conceptual Model (ACM), contextual conceptual user model (CCM) and augmented contextual conceptual user model (ACCM) compared to related works, we can observe the efficiency of augmented contextual conceptual model which achieves the best among all other models, Entity model achieves moderate efficiency, whereas using Steiner tree to extend the model shows a good effect on conceptual user model and should be investigated more in the future (Figure 6).

Table 5. Comparison of different models with different measures

<table>
<thead>
<tr>
<th></th>
<th>MRR</th>
<th>P@5</th>
<th>P@10</th>
<th>S@5</th>
<th>S@10</th>
<th>R@10</th>
<th>F1@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>0.294474</td>
<td>0.171429</td>
<td>0.153571</td>
<td>0.379464</td>
<td>0.522321</td>
<td>0.008807</td>
<td>0.016577</td>
</tr>
<tr>
<td>Tf-idf</td>
<td>0.477271</td>
<td>0.360714</td>
<td>0.345536</td>
<td>0.562500</td>
<td>0.678571</td>
<td>0.019895</td>
<td>0.037452</td>
</tr>
<tr>
<td>PM</td>
<td>0.530825</td>
<td>0.413393</td>
<td>0.380804</td>
<td>0.660714</td>
<td>0.754464</td>
<td>0.021749</td>
<td>0.040960</td>
</tr>
<tr>
<td>EM</td>
<td>0.627313</td>
<td>0.475893</td>
<td>0.436607</td>
<td>0.763393</td>
<td>0.848214</td>
<td>0.025495</td>
<td>0.047925</td>
</tr>
<tr>
<td>STM</td>
<td>0.714115</td>
<td>0.561607</td>
<td>0.528125</td>
<td>0.861607</td>
<td>0.915179</td>
<td>0.030327</td>
<td>0.057061</td>
</tr>
<tr>
<td>CM</td>
<td>0.693175</td>
<td>0.541071</td>
<td>0.516518</td>
<td>0.834821</td>
<td>0.901786</td>
<td>0.029725</td>
<td>0.055917</td>
</tr>
<tr>
<td>ACM</td>
<td>0.730058</td>
<td>0.558929</td>
<td>0.523214</td>
<td>0.866071</td>
<td>0.915179</td>
<td>0.030156</td>
<td>0.056722</td>
</tr>
<tr>
<td>CCM</td>
<td>0.816321</td>
<td>0.653571</td>
<td>0.604018</td>
<td>0.933036</td>
<td>0.964286</td>
<td>0.034852</td>
<td>0.065557</td>
</tr>
<tr>
<td>ACCM</td>
<td>0.823518</td>
<td>0.657143</td>
<td>0.609375</td>
<td>0.928571</td>
<td>0.96875</td>
<td>0.035206</td>
<td>0.066218</td>
</tr>
</tbody>
</table>

Figure 6. MRR value for different methods
CONCLUSION

It is important to build personalized recommendation systems for online social networks, so in this paper we introduced a new approach to building a contextual and conceptual user models and proposed a recommendation algorithms based on these models, we showed a method to represent text of user items as sequence of concepts extracted from a knowledge base and presented a new method to exploit the semantic relation of the knowledge base to expand user model that proves good results, we compared our models and algorithms with most state of the art works and showed how our models outperform all of them, our experiment proved that the conceptual representation of the text is more effective than using named-entities, besides, the inclusion of contextual information in the conceptual representation shows a very promising results, so that this orientation should be investigated using more tools in the future.
REFERENCES


Ammar Alnahhas is a teacher at Damascus University, his interests include natural language processing, knowledge management and machine learning.

Bassel Alkhatib is an associate professor for Damascus University.