Towards Recommendation Using Interest-Based Communities in Attributed Social Networks

Amani H. B. Eissa Mohamed E. El-Sharkawi Hoda M. O. Mokhtar
(a.hassan, m.elsharkawi, h.mokhtar)@fci-cu.edu.eg
Department of Information Systems
Faculty of Computers and Information
Cairo University, Cairo, Egypt

ABSTRACT

Social networks can be modeled as attributed networks whose nodes represent users, edges represent relationships among users (e.g. friendship/follow) and attribute vectors hold properties of nodes and/or edges. In this paper, we consider friends' recommendation based on interest-based communities generated from topic based attributed social networks (TbASN). In our model, an attribute vector is not just a container for explicit users' profile data that is stored in social network's dataset, but rather holds topic vectors that are derived from analyzing the implicit interest of users' that are aggregated from his/her posts on the social network (e.g. tweets in Twitter, posts in Facebook). In our framework, topics of interest are represented as a hierarchy of topics (Topics/Subtopics) forming hierarchical interest-based communities. Users within each interest-based community are clustered according to their profile features (age, location, education… etc.). Those clusters are later used in recommendations where recommendations target members of the same cluster to guarantee the quality and coherence of recommendations. In addition, we propose a recommendation selection approach to handle the large number of recommended candidates. The main advantage of the proposed approach is that it considers multiple criteria for candidate selection including the number of common communities, the resemblance in basic features, as well as network proximity.

In addition to recommending friends of similar interests, frequent pattern mining is used to discover frequently occurring interests in order to be used in recommending communities for users to join. Although our approach is generic and can be applied to most of the existing social networks, we used Twitter as our target social network.

KEYWORDS

social networks, attributed networks, recommendation, community detection, topic identification

WWW '18 Companion, April 23 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04
https://doi.org/10.1145/3184558.3191562

ACM Reference format:

Amani H. B. Eissa, Mohamed E. El-Sharkawi and Hoda M. O. Mokhtar. 2018. Towards Recommendation Using Interest-Based Communities in Attributed Social Networks. In *Proceedings of The 2018 Web Conference Companion (WWW '18 Companion). ACM, New York, NY, USA, 8 pages.* https://doi.org/10.1145/3184558.3191562

1 INTRODUCTION

A social network (SN) is represented as a graph, where nodes are users and edges are the relationships among users. Edges may be either directed (like follows relationship in Twitter or Instagram) or undirected (like friendship in Facebook or co-authorship in DBLP). Attributed social networks, ASN, have emerged recently as an extension of social networks (SN). In attributed social networks [6], the basic graph structure of the network is extended by augmenting nodes and/or edges with attributes that carry more information about nodes/edges, and hence allow further and more useful analysis of the network.

In this paper, the users of the social network are augmented with their interests. That is, each user has a vector the holds a list of her topics of interest which are deduced from user's posts. A topic tree is constructed using the available knowledge base DMOZ [4][8]. Nodes of the topic tree are extended with keywords related to the node's topic. Those keywords are extracted from users' posts and other external sources like DBpedia [3][10] and WordNet [12][11]. We exploit this information to construct virtual interest based communities (IBC) such that all members in a IBC share the same interest but not necessarily are topologically related, for example, community of users who are interested in the topic classic music, accordingly be members of a community "Classic Music", however, they are not directly connected by a follow link. As topics are constructed in a hierarchical topic/subtopic structure, nested communities are built where a member of the "Classic Music" community is consequently a member of the super community "Music".

Exploiting the virtual interest communities, we recommend friends to users based on common interest. Friends recommended to a user do not necessarily have a transitive edge-based relationship with the user (i.e. no common friends between the user and the recommended friend in the network). In order to

generate homogeneous recommendations, we cluster communities based on basic attributes of users. Users in the same cluster are candidates for recommendation. For instance, we may cluster the "Classis Music" community on the basic attribute "Age", and accordingly, members in the same age group are candidate for recommendation. Hence, our proposed recommendation approach is not only based on common interests among users but also common profile attributes that they share to guarantee coherent recommendations.

Contributions of this paper are three fold; (1) describing an approach to augment members of the social network with their interests; (2) building hierarchies of virtual communities based on users' interests; and (3) producing friends' recommendations to interest based communities' members. We consider Twitter social network to apply and describe our approach.

The paper is organized as follows. In the next section we review related work on topic identification as well as community detection and recommendation in attributed networks. Section 3 illustrates the interest based communities generation process. The friends' recommendation based on IBCs is explained in Section 4. Section 5 describes the first steps and datasets used to realize the proposed framework. Finally Section 6 concludes our work and states some directions for future work.

2 RELATED WORK

In this section we review work related to main concepts used in our framework; i.e. topic identification, community detection and recommendation in ASN.

Identifying the topics of user posts within social networks is the interest of many research and utilized in many applications [16, 21, 24, 26]. Many studies also focused on building ontologies for user interests like [10, 19]. The topic tree we use in this paper could be compared to [10] which focuses on building ontologies for users interests using multiple knowledge bases. In [10], the authors used latent semantic analysis technique to measure the similarity between interests based on their definitions on knowledge bases like Wikipedia [28], WordNet and DMoz. They also applied clustering to group similar interests together into higher level concepts. Inspired by their results, which states that DMoz best represents social network user interests over other knowledge bases, the tree we used (described in detail in [5]) is built mainly using the first two levels of DMoz hierarchy. Moreover, we augment the tree nodes with topic-related keywords aggregated from multiple sources to help in topic identification.

Representing social networks as ASN has been used in many studies like [7, 8, 12–15]. Most of these studies store node or edge attributes derived from the explicit basic data for users/relationships. Very few studies augment the attributes of the network with data derived from user posted contents and/or user interactions in the network. The work in [7] is among those who consider adding keywords, that are derived from the user's interests, to the node attributes to be used in answering community queries within the network, however, the method of retrieving those keywords was not explained in the paper. In our approach, a detailed method of associating users with their implicit interests derived from their posts in the network is described.

Moreover, the area of community detection CD has always been a core part of SN research. Many studies like [25, 27, 29, 30] discussed CD in attributed networks. The majority of the CD approaches rely mainly of the topological structure of the network of detect communities, some mix the topological structure with the attribute similarity like [7, 15, 18]. Work in [20] suggests CD using deep learning to determine which dimension in user profiles leads to community cohesiveness. On the other hand, our approach generates virtual communities that mainly preserve resemblance of both users' implicit interests as well as explicit basic feature regardless of network connectivity.

Many studies have addressed recommendation in social networks like [6, 31]. Our approach for friends' recommendation considers sharing multiple interests, similar basic features and degree on interest factor to select high quality recommendations.

3 GENERATING INTEREST-BASED COMMUNITIES (IBCS) FROM ASN

In this section, we present the procedure for generating Twitter users' virtual communities that do not rely on the topological structure of the network, but rather on users' common interests. Section 3.1 represents the twitter network as a topic-based attributed social network along with the approach to assign topics of interest to users. In Section 3.2, virtual interest-based communities are generated using implicit users' interests that are driven from users' interactions on Twitter.

3.1 Topic based Attributed Network Representation for Twitter

Representing Twitter as an ASN is achieved by augmenting nodes with two attribute vectors: *user features vector (UFV)* and *user interests vector (UIV)*. The user features vector (UFV) is derived from users' basic data that are explicit in their profiles that are stored in social network's data store. These features include: date of birth/age, location at different granularities (district, city, Country), education, profession, etc.

The second vector that is attached to each node description is called: *user interests vector (UIV)* which holds user's topics of interests implicitly obtained from his/her tweets. The following

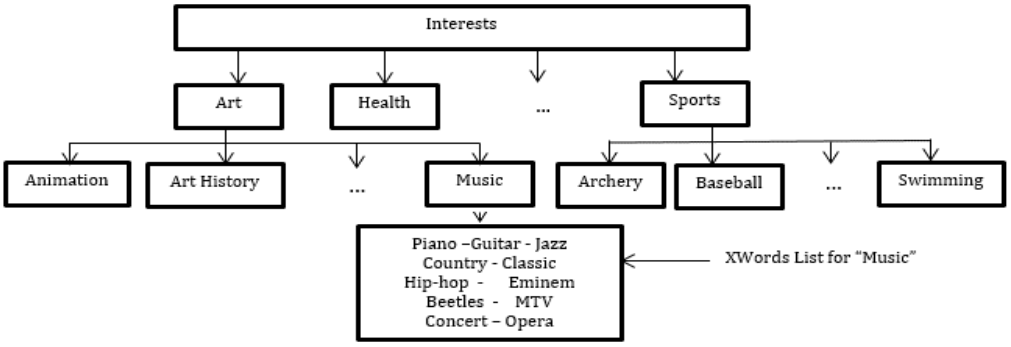

Figure 1: An example of 2-levels interests topic tree with XWords lists

subsections introduce the process to construct UIVs. UIV construction is basically done in two steps: first identifying tweet topics (described in 3.1.1), and then, generating UIVs (discussed in 3.1.2)

3.1.1 Identifying tweet topics:

To identify topics within user tweets, a topic tree is employed as suggested in [5], where the topics are organized in a topics/subtopics hierarchical structure, and each leaf node of the tree is associated with a sorted list of topic related keywords called the "XWords list". These keywords are initially extracted from multiple sources including DMoz, DBpedia, and WordNet. The XWords of an inner node of the topic tree is the union of the XWords of the nodes of its descendants. Figure1 shows an example of a 2-level topic tree**.** The XWords list for the node Music (subtopic of Arts) contains the keywords {Piano, Bands, Guitar, Jazz, Lyrics, Country, Classic, Hip Hop, Eminem, Beetles, MTV, Concert, Opera, iPod, speakers, headphones, etc.}. XWords lists are extended by augmenting the XWords generated in [5] by keywords extracted from the tweets themselves. This aggregation of topics is achieved through performing co-occurrence analysis to extract the words that frequently co-occur with the topics' XWords. For example, the hashtag "#Rio2016" was found to co-occur in a large number of tweets that contain the words "sports, swimming, triathlon, stadium, gymnastics… etc.", those tweet keywords are all members for the XWords list of the node "Sports". Therefore, the word "#Rio2016" is consequently added to the XWords list for the node "Sports". Nevertheless, XWords are also used to identify a topic in case the tweet does not explicitly contain the topic's name. For example: "200M butterfly is a tough race", this tweet does not explicitly mention the topic "swimming", however, the words {butterfly, race} are found in "swimming" XWords list hence enable associating the topic "swimming" to the tweet's users topics of interest.

The steps for topic assignment to tweets are elaborated in Algorithm 1. The input to the algorithm is a set of "clean" tweets and the topic tree. Tweet cleaning is performed as a preprocessing step in which all non-English tweets are excluded and all stop words are removed from the tweets' text. For each user, the algorithm selects all her tweets, sorts the clean tweet words in an alphabetical order to facilitate the search, then maps each word to the corresponding topic using the tree's XWords lists.

Let $TWEETS = \{tw_1, tw_2,\ldots,tw_n\}$ be the set of clean tweets, each tweet tw_i is a triplet $(TID_i, wordsList_i, UID)$ where TID_i is the tweet ID, $wordsList_i$ is the list of words in tw_i and UID is the user who posted the tweet. Similarly, let $TopicTree\ \{(t_1, XList_1), (t_2, XList_2),\ldots(t_m, XList_m)\}$ be the topic tree represented as a set of $(t_j, XList_j)$ pairs where t_j is a topic name of a leaf node and $XList_j$ is the set of indicating keywords XWords for t_j. $topicList(tw_i)$ is the list of topics for tweet tw_i such that if a word w in tw_i is member of any topic t_j of the *TopicTree*, t_j will be added to the $topicList(tw_i)$.

The proposed topic assignment algorithm proceeds as follows: First, the algorithm examines each word $\boldsymbol{w}$ in the *wordsList* of each clean tweet, if $\boldsymbol{w}$ is explicitly a topic in the topic tree, it adds the topic to the tweet's topic list. Otherwise, it checks the existence of $\boldsymbol{w}$ in all XWords lists in the tree, if $\boldsymbol{w}$ is found in one of the XWords lists, the list's topic is added to the tweet's topic list. In case that the word $\boldsymbol{w}$ exists in more than one XWords list (this means that this word may indicate more than one topic), all matching topics are added to the topic list of the tweet. After examining all the tweet's words, the following cases are examined:

Case1: None of the tweet words map to any topic the tweet's topic is marked to be "undefined".

Case 2: A tweet is mapped to multiple topics we count the number of occurrences of each topic in the tweet's topic list, using a majority function, the topic having the maximum of occurrences is considered to be the tweet's topic.

Case 3: A tweet is mapped to multiple topics with equal frequencies In case that the tweet is mapped to multiple topics and none of these topic get the majority of occurrences (i.e. the topic appears with equal frequency in multiple topics), in this

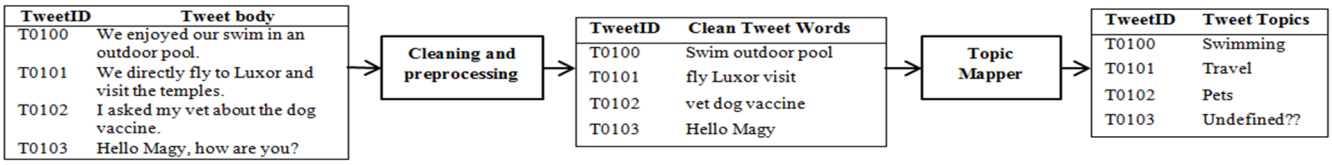

Figure 2: Mapping tweets to topics

case, we investigate whether some (or all) of these topics have a common parent topic, then the parent topic is assigned to be the tweet's topic. For example, in the topic list {gymnastics, karate, swimming, music}, none to the topics gets a majority of occurrence, however, generalizing the first three topics {gymnastics, karate, swimming} to their parent topic: "Sports" an overall frequency of occurrence of 75% is achieved. Nevertheless, if generalizing topics in the topic list is not possible to break the tie among equally weighted topics, i.e. a common parent is not found, then, the topic of the tweet is considered "undefined".

Fig. 2 illustrates an example for mapping tweets to topics. First, data cleaning removed all stop words, then topic mapper matches clean tweet words to their corresponding topics by looking up topics' XWords.

3.1.2 Generating User Interests Vectors (UIVs):

User interests vectors (UIVs) are constructed by analyzing user posts on the network. Studying users' interactions over social networks makes it easy to deduce that different users have different degrees of interest in different topics. Although this degree might not be expressed explicitly in the user profile or even mentioned in her interest list, yet analyzing users' tweets and the topics of those tweets provides a way to infer the user's degree of interest in each topic. For the rest of this discussion, let DIF represents a measure of the degree of interest of a user in a specific topic ***t***. Items in a user interest vector are triplets (t_i, $count_i$, DIF_i) which mean that the user is interested in topic t_i that appeared with frequency $count_i$ in her tweets and with degree of interest *DIF*.

Given that we already know the topics of the user's tweets via applying Algorithm 1, we can now aggregate them in one interest vector for each user. To obtain meaningful results, a user is not considered to be interested in a topic unless the topic has been mentioned a minimum number of occurrences in the user's tweets. To do that, a fixed minimum threshold is defined to determine whether a topic is of interest to a user or not. However, users' activity level within the social network is not constant, for instance if we set a minimum threshold of 5 times/month for example to mark the user as "interested" in a topic, it might be meaningful for a user who posts 10 tweets per month, and should be extremely trivial for an active user who posts 100 tweets per day. Therefore, predefining the threshold value and fixing it to all users is meaningless, thus, instead of setting a global threshold

Algorithm1: Topic Assignment to tweets

```
1  Input: TWEETS, TopicTree
2  Output: topicList for every tweet in TWEETS
3  Begin:
4  for each twi in TWEETS do:
5    for each word w in wordsListi do:
6      if w is equal to any tj ∈ TopicTree then: // word is
7          add tj to topicList(twi);            // a topic name
8      else:
9          for each list XListv in TopicTree do:
10             if w exists in XListv then:
11                 add tv to topicList(twi);
12 if |topicList(twi)| = 0 then:
13    return "undefined";
14 for each topic tx in topicList(twi) do:
15   if |tx| in topicList(twi) ≥ |topicList(twi)|/2 then:
16     return tx ;
17   else if (getCommonParent (topicList (twi)) = true) then:
18     return commonParent(topicList (twi);
19   else return "undefined";
20 End;
```

that applies to all users, a minimum threshold is defined for each user based on the percentage of her total number of posts. For example, we may assume that a user is interested in a topic if she tweets about it more than 20% for her tweets during the period of analysis.

Additionally, for each topic of interest for each user, a measure called Degree of Interest Factor (DIF) is calculated. The DIF of user u in topic t is calculated by dividing the number of tweets the user u posted in topic t by the total number of tweets posted by u multiplied by the number of topics in u's interest vector.

$$\mathrm{DIF}(u,t) = \frac{\text{Number of tweets u posted about t}}{\text{total number of tweets u posted}} * \text{Number of topics in u's interest Vector} \quad (1)$$

The DIF measure is an indication of how significant topic t is to user u with respect to the number of topics of interest to user. For instance, consider the user C who has 5 topics in his UIV, C tweeted a total of 100 tweets, 10 of which in topic t1. While, another user D who has 50 topics in his UIV, also tweeted a total of 100 tweets, 10 of which in topic t2. Despite the similarity of the total number of tweets and the number of tweets in a given topic, the DIF value in both cases is not the same. In the first case, DIF

(C,t1) is calculated to be 0.5 which means that user C is interested in t1, half of the value he would be if he was equally interested in all of the topics in his UIV; i.e. C would tweet 20 times in each topic if he is equally interested in all of the topics in his UIV. While the DIF (D,t2) is calculated to be 5, which means that the user D is interested in t2 five times more than he would, if he is interested in all of the topics in his UIV equally. Hence, the bigger the value of DIF (u,t), the higher the significance of topic t to user u. It is also worth to mention that the DIF measure is not affected by the user activity level on the social network as it is taking into consideration the total number of the user's tweets as well as the number of topics in the UIV.

To construct the users' UIVs , assume U as the set of users, L_{tt} (TID, UID, t) as the tweet/topics lists for each user, where *TID* is the tweet ID, *UID* is the user ID of the user who posted the tweet and *t* is the tweet topic identified by algorithm1. A user *u* is considered interested in topic *t* if the number of tweets written in topic *t* by *u* divided by the number of all tweets written by *u* is greater than or equal minimum threshold *min_threshold*. Algorithm 2 constructs *U*ser *I*nterest *V*ector (*UIV*) for each user *u* ϵ U. *UIV(u)* is the vector of triplets (t, count, DIF) for user *u*. The inputs of Algorithm 2 are: **U** List of users and $\mathbf{L_{tt}}$ tweet/topic list for each user and **min_threshold.**

Algorithm2: Construct User Interests Vectors with DIFs

```
1  Inputs: Ltt, U, min_threshold
2  Output: UIV for each user u ϵ U
3  Begin:
4  for each user u in U do:
5    initiate a list UIV(u) (topic, count, DIF)= ϕ;
6      Select Ltt (TID,UID,t), count(tweets) from Ltt where
       UID = u AS UserTweets, NumTweets group by UID;
7  for each tweet tw in UserTweets do:
8    if t do not exist in UIV(u) then:
9      add (t,1) to UIV(u);
10   else:
11     modify (t, count) to (t, count+1);
12 threshold = min_threshold * UserTweetsCount;
13 remove from UIV(u) all topics with count < threshold;
14 for each topic t in UIV(u) do:
15     DIF(t) = count(t)*count (UIV(u))/NumTweets;
16 return UIV(u);
17 End;
```

3.2 Generating Interest based Communities (IBCs) from user interest vectors UIVs

Having now the interest vector for each user, we can accordingly construct *virtual interest based communities (IBCs)*. By virtual we mean that members of each interest based community are not necessarily topologically connected, however, they are said to belong to the same community depending solely on their common interests.

We start by traversing the leaf nodes of the topic tree, we create a *IBCsList*, which is a set of *(t, L_{users})*, where *t* is a topic name at the deepest level of the *TopicTree*, L_{users} is a list of users interested in topic *tp*. L_{users} lists are filled with all users who have the topic *t* in their *UIVs*. Algorithm 3 describes the interest based communities construction process. The inputs to the algorithm are users *UIVs* and the *TopicTree*.

Dealing with the hierarchical nature of the topic tree enables the generation of interest based communities at multiple degrees of granularity. For example, "swimming" and "water polo" communities can be generalized into "water sports" community which can be in turn generalized into "Sports" community.

Algorithm3: Construct Interest Based Communities

```
1  Input: Users U, UIVs, TopicTree
2  Output: IBCsList
3  Begin:
4  for each topic t ∈ leaf nodes of TopicTree do:
5    Initiate IBCsList (t);
6  for each user u in U do:
7    for each t in UIV(u) then:
8      add u to IBCsList (t). Lusers;
9  return IBCsList;
10 End;
```

4 GENERATING RECOMMENDATIONS USING INTEREST BASED COMMUNITIES

Many types of recommendations can be generated to users of the social network including: friends, hashtags, events and places. In this section, we focus on generating interest-based friend recommendations for IBC members.

Applying the concept of *homophily* which describes the tendency of individuals to associate and bond with similar others, as in the proverb "birds of a feather flock together", we build the friend recommendations upon both the co-membership in identified IBCs and the similarity in attributes in UFVs.

Given a user X who is a member of communities: $C_1, C_2, \ldots, C_n$. We need to generate interest based friends' recommendation to X based on her interests in topics $t_1, t_2, \ldots, t_n$. Recommending friends within the same communities for X faces the following challenges:

1. Some communities may contain a huge number of recommendation candidates. For example: Liverpool soccer club fans community can contain one million members.

2. Heterogeneous characteristics of users within the same community. For instance, soccer fans include all age groups, considering only "Soccer" as a common interest may lead to

recommending a 15-year-old high schooler John to Prof. Matthew (a 60-year-old full professor).

3. As the number of candidate friends for recommendations is large, ranking criteria are essential in order to recommend "best" matching friends.

Therefore, our approach considers the following refinements to produce high quality and relevant recommendations:

1. Applying feature based filtering: Within each *IBC*, community members are clustered based on their *UFV* attributes (i.e. age, education, location, income, occupation, etc.}. After the clusters are formed recommending friends from the same cluster within the same *IBC* produces more coherent results. Referring to the example above, by clustering the "soccer" community based on the age attribute, John should be offered friends recommendations from the cluster "teenagers" within the soccer community.

2. Considering community overlapping: as *IBCs* overlap, recommending friends who share more than one community with a user is definitely better than recommendations based on only one common interest. For user *X* who is a member of communities $C_1, C_2, \ldots C_n$, we recommend for *X* a list of friends *R(X)*, where members in *R(X)* belong to the maximal intersection of all communities containing *X*. The larger the number of overlapped communities the more interesting the recommendation is.

$$R(X) = \{f : f \in \bigcap_{i=1}^{n} C_i\} \quad (2)$$

3. Performing frequent pattern mining on UIVs to detect frequent users common interests: Applying frequent pattern mining [7] on UIVs detect frequently occurring interests. Frequent patterns of interests actually represent community overlapping. It may sound quite straightforward to find a frequent pattern stating that people who are interested in *diving* are also interested in *water polo* because both topics belong to the same parent category "water sports", however, a more interesting finding is to obtain a frequent pattern stating that: people who are interested in *running* are also interested in *music* because both topics belong to two different parent categories. Therefore, recommendations based on frequent patterns of interests that do not belong to the same direct common parent are likely to be more interesting and offer more diversity to users.

To address the challenges of selecting candidates for recommendation for a given user, a selection approach is proposed which applies consecutive filters to choose the best recommendation candidates. The selection approach steps are described as follows: *Let X* be the user who is the target for recommendation, *Comm(X)* is the set of communities $\{C_1, C_2, \ldots C_n\}$ where *X* is a member of all communities in *Comm(X)*, *UIV(X)* is the interests vector of user *X*, *UFV(X)* is the vector which stores a list of (attribute, value) pairs for user *X*, i.e. age group, gender, marital status, level of education and location. *Clusters(X)= {CL$_i$}* is the set of clusters for user *X*, where user *X* belongs to the cluster *CL* within community C_i where $C_i \in Comm(X)$. *R(X)* is the set of candidates for recommendations for *X*. maxCandidates is a parameter used to set the maximum number of friends to be recommended to *X*. β is a parameter used to set the maximum acceptable degree of interest difference between the user *X* and each candidate friend $f \in R(X)$ with respect to a topic.

Step 1: get the set of candidate friends *F* who belong to maximum intersection of all communities in *Comm (X)*, add $\forall f \in F$ to *R(X)*.

Step 2: generate Clusters (*X*), with respect to each community in *Comm(X)*, in which all members of *Cluster(X)* have similar *UFVs* as *X*.

Step 3: Consider only users in *R(X)* that are also members in Clusters (*X*), i.e. $R(X) = R(X) \cap Clusters(X)$.

Step 4: Refine *R(X)* by keeping the recommendation candidates who have matching degree of interest DIF_i in a topic t_i, corresponding to each community C_i, as *X*.

Step 5**:** if, after all previous refinement steps, the number of candidates for recommendation is still large, we propose applying a network proximity ranking. For every candidate f ∈ R(X), computes shortest path (X, f), sort R(X) based on shortest path ascendingly i.e. recommend f ∈ R(X) where f has minimal shortest path with X first.

Algorithm4: Generate friends recommendation for a user X

```
1   Input: X, Comm(X), Clusters(X), UIV(X), UFV(X),
    maxCandidates, β
2   Output: R(X);
3   Begin:
4   initiate list R(X) = Φ;
5   R(X) = get candidates F: f ∈ ⋂_{i=1}^{n} C_i where C_i ∈ Comm(X);
6   if |R(X)| ≤ maxCandidates then:
7     return R(X);
8   generate Clusters (X, Comm(X));
9   R(X) = R(X) ∩ Clusters(X);
10  if |R(X)| ≤ maxCandidates then:
11    return R(X);
12  for each user f in R(X) do:
13      if |DIF(f, t_i) - DIF(X, t_i)| ≥ β then:
14          R(X)=R(X)-f;
15  if |R(X)| ≤ maxCandidates then:
16    return R(X);
17  for each user f in R(X) do:
18     calculate SP(U) = ShortestPath(f, X);
19  sort R(X) w.r.t SP(f) ascendingly;
20  return R(X);
21  End;
```

5 IMPLEMENTATION AND DATASETS

To validate our proposed approach, we are currently implementing a prototype using a Twitter data set that contains randomly extracted tweets and their users. It comprises around 180M (exactly 180400196) tweets for around 263k (exactly 262989) users over 5 months period from October 2015 to February 2016, which is an extended dataset of the one used in [9].

We built an interests topic tree by extracting selected 7 categories out of 16 categories contained the online DMoz [4] to stand as the first level of our topic tree. The chosen categories are {Arts, Health, News, Recreation, Science, Shopping and Sports}. Second, we retrieved the second level of the DMoz hierarchy and manually edited it to exclude irrelevant nodes, i.e. nodes that do not describe "interest" topics (e.g. A, advice, Africa … etc.). Next, for each node/topic, we extracted relevant keywords from DBpedia, WordNet, and Amazon Categories tree data [1] via Amazon web services (AWS) [2], respectively, and retrieve sets of keywords and add them to XWords list for each node in the tree. As a refinement step, we augmented the XWords lists of the tree by performing a co-occurrence analysis on the tweets dataset to extract the words which frequently occurred topic names within the tweets text. The extracted frequent words are added to the corresponding XWords lists.

At the moment, converting the Twitter data into the ASN format is underway, by first mapping tweets to corresponding topic and then constructing both user features and interest vectors. Next, the IBCs will be generated and the friend recommendation module will be implemented.

6 CONCLUSION

In this paper we presented an attributed social network representation for Twitter, where users are augmented with interests implicitly extracted from their tweet contents. For each tweet, we identify its topic(s) using a hierarchical interests' topic tree. Based on user interests, we construct interest based communities (IBCs) where all users in a community share interest in the same topic.

The second contribution of the paper is friend recommendation using IBCs, where the friends recommended to a user share similar interests and similar features. To generate high quality, homophily preserving, recommendations, we propose four filtering criteria that reduce the number of recommendation candidates.

Currently, we are implementing the proposed approach and investigating possible techniques to optimize its performance (e.g. using different indexing structures to help in topic identification).

As future work, capturing temporary interests throughout our approach may be considered. Temporary interests phenomenon arises when a user posts about a certain topic intensively, although the topic is not in the user's UIV, during a certain time interval and then stops shortly later on. Temporary interests may represent a sudden shift in users' attention due to the special nature of the topic itself. For example, during elections almost everyone speaks about politics, during Football World Cup or Olympics many people post about sports. By adding the time dimension to our analysis to detect such phenomenon, we may exploit adding an additional attribute the user interest vector to mark some topics as temporary. By mining user UIVs across the SN, common temporary interests captured on large scale in the network may be an approach to event detection. Moreover, studying the evolution of user interests, and accordingly interest-based communities in ASN is a promising direction for future research.

REFERENCES

[1] Amazon Categories Tree: *http://amazoncategories.info/.*

[2] Amazon Web Services: *https://aws.amazon.com/.*

[3] Auer, S. et al. 2007. DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web* (Berlin, Heidelberg, 2007), 722–735.

[4] DMOZ directory database: *https://dmoztools.net/docs/en/rdf.html.*

[5] Eissa, A.H.B. et al. 2016. Constructing Topic Hierarchy Tree for User Interests From Multiple Knowledge Bases. *The Fifth International Conference on Informatics and Applications (ICIA2016)* (2016), 123–126.

[6] Epasto, A. et al. 2015. Ego-net Community Mining Applied to Friend Suggestion. *Proceedings of the VLDB Endowment* . 9, 4 (2015), 324–335. DOI:https://doi.org/10.14778/2856318.2856327.

[7] Fang, Y. et al. 2016. Effective Community Search for Large Attributed Graphs. *Proceedings of the VLDB Endowment.* 9, 12 (2016), 1233–1244. DOI:https://doi.org/10.14778/2994509.2994538.

[8] Gong, N.Z. et al. 2012. Evolution of Social-Attribute Networks: Measurements, Modeling, and Implications using Google+. *CoRR.* abs/1209.0, (2012).

[9]Han, J. et al. 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery.* 15, 1 (Aug. 2007), 55–86. DOI:https://doi.org/10.1007/s10618-006-0059-1.

[10] Haridas, M. and Caragea, D. 2009. Exploring Wikipedia and DMoz as Knowledge Bases for Engineering a User Interests Hierarchy for Social Network Applications. *On the Move to Meaningful Internet Systems: OTM 2009* (Berlin, Heidelberg, 2009), 1238–1245.

[11] Hassan, N. et al. 2016. Measuring User's Susceptibility to Influence in Twitter. Social Data Analytics and Management Workshop, co-located with VLDB 2016 (2016).

[12] Huang, Y. and Wangg, H. 2017. Consensus and multiplex approach for community detection in attributed networks. *2016 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2016 - Proceedings.* (2017), 425–429. DOI:https://doi.org/10.1109/GlobalSIP.2016.7905877.

[13] Jia, C. et al. 2017. Node Attribute-enhanced Community Detection in Complex Networks. *Scientific Reports.* 7, 1 (2017), 1–15. DOI:https://doi.org/10.1038/s41598-017-02751-8.

[14] Kim, M. and Leskovec, J. 2011. Modeling social networks with node attributes using the multiplicative attribute graph model. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI. 400-409,* (2011), 33. DOI:https://doi.org/10.1080/15427951.2012.625257.

[15] Largeron, C. et al. 2015. Generating attributed networks with communities. *PLoS ONE.* 10, 4 (2015), 1–13. DOI:https://doi.org/10.1371/journal.pone.0122777.

[16] Lee, K. et al. 2011. Twitter trending topic classification. *Proceedings - IEEE International Conference on Data Mining, ICDM.* (2011), 251–258. DOI:https://doi.org/10.1109/ICDMW.2011.171.

[17] Lehmann, J. et al. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*. 6, 2 (2015), 167–195. DOI:https://doi.org/10.3233/SW-140134.

[18] Liao, L. et al. 2017. Attributed Social Network Embedding. 14, 8 (2017), 1–12.

[19] Ma, Z. et al. 2005. Evaluation of Ontology-based User Interests Modeling. *Proceedings of the 4Th Workshop on E-Business* (2005).

[20] McAuley, J. and Leskovec, J. 2012. Discovering Social Circles in Ego Networks. ACM Trans. Knowl. Discov. Data 8, 1 (2012). DOI:https://doi.org/10.1145/2556612.

[21] Michelson, M. and Macskassy, S.A. 2010. Discovering users' topics of interest on twitter. *Proceedings of the fourth workshop on Analytics for noisy unstructured text data - AND '10*. (2010), 73. DOI:https://doi.org/10.1145/1871840.1871852.

[22] Miller, G.A. 1995. WordNet: a lexical database for English. *Communications of the ACM*. 38, 11 (1995), 39–41. DOI:https://doi.org/10.1145/219717.219748.

[23] Miller, G.A. et al. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*. 3, 4 (1990), 235–244.

[24] Park, S. and Shin, H. 2014. Identification of Implicit Topics in Twitter Data Not Containing Explicit Search Queries. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. March 2013 (2014), 58–68.

[25] Ruan, Y. et al. 2013. Efficient community detection in large networks using content and links. (2013),. Proceesing of the 22nd international conference on World Wide Web. 1089–1098 DOI:https://doi.org/10.1109/BRICS-CCI-CBIC.2013.117.

[26] Sokolova, M. et al. 2016. Topic Modelling and Event Identification from Twitter Textual Data. CoRR, abs/1608.02519 , (2016), 17.

[27] Wang, X. et al. 2017. Community detection in attributed networks based on heterogeneous vertex interactions. *Applied Intelligence*. 47, 4 (2017), 1270–1281. DOI:https://doi.org/10.1007/s10489-017-0948-6.

[28] Wikipedia, The Free Encyclopedia: *https://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350*. Accessed: 2004-07-22.

[29] Xu, Z. et al. 2012. A model-based approach to attributed graph clustering. *Proceedings of the 2012 international conference on Management of Data - SIGMOD '12*. (2012), 505. DOI:https://doi.org/10.1145/2213836.2213894.

[30] Yang, J. et al. 2013. Community detection in networks with node attributes. *Proceedings - IEEE International Conference on Data Mining, ICDM*. (2013), 1151–1156. DOI:https://doi.org/10.1109/ICDM.2013.167.

[31] Yi, J. et al. 2016. Incorporating Multiple Attributes in Social Networks to Enhance the Collaborative Filtering Recommendation Algorithm. 7, 4 International Journal of Advanced Computer Science and Applications. 7. . 10.14569/IJACSA.2016.070408. (2016), 60–67.