Predicting Future Interests in a Research Paper Recommender System using a Community Centric Tree of Concepts Model

Modhi Al Alshaikh, Gulden Uchyigit and Roger Evans
School of Computing, Engineering and Mathematics, University of Brighton, Brighton, U.K.

Keywords: Recommender Systems, Collaborative Filtering, Information Retrieval, Research Paper Recommendations.

Abstract: Our goal in this paper is to predict a user's future interests in the research paper domain. Content-based recommender systems can recommend a set of papers that relate to a user's current interests. However, they may not be able to predict a user's future interests. Collaborative filtering approaches may predict a user's future interests for movies, music or e-commerce domains. However, existing collaborative filtering approaches are not appropriate for the research paper domain, because they depend on large numbers of user ratings which are not available in the research paper domain. In this paper, we present a novel collaborative filtering method that does not depend on user ratings. Our novel method computes the similarity between users according to user profiles which are represented using the dynamic normalized tree of concepts model using the 2012 ACM Computing Classification System (CCS) ontology. Further, a community-centric tree of concepts is generated and used to make recommendations. Offline evaluations are performed using the BibSonomy dataset. Our model is compared with two baselines. The results show that our model significantly outperforms the two baselines and avoids the problem of sparsity.

1 INTRODUCTION

Most research paper recommender systems suggest research papers which are similar to a user's profile which result in a limited set of recommendations based on current user preferences that are represented in the system (Kotkov et al., 2016). A major challenge in recommender systems is to explore the potential of future interests of users (Yang et al., 2016). Content-based approaches are able to recommend a set of papers that relate to user's current interests. However, they suffer from the problem of content overspecialization because they depend only on the metadata of papers in the user's profile; therefore the user is restricted to getting recommendations similar to papers already defined in his/her profile (Isinkaye et al., 2015). Collaborative filtering approaches have the ability to explore potential future interests. Existing collaborative approaches have been developed for domains such as movies, music and e-commerce products. These collaborative approaches are not appropriate for the research paper domain, because they depend on large numbers of user ratings. However, there is a lack of ratings in the research paper domain (Yang et al. 2009). For example, the implicit ratings (users' access logs) on Mendeley[1] (research paper domain) has been compared with Netflix[2] (movie domain), has been found that the sparsity of Mendeley was three orders of magnitude higher than on Netflix (Beel et al., 2016). This is due to the different behaviour of users in these two domains. For example in the movie domain there are many users who have watched the same movies. Therefore, similar users can be found for most users and hence recommendations can be made effectively. However, the research paper domain suffers from the data sparsity problem, where several new papers have not been read by any user and further, a new user may read only a few papers (Jain 2012; Beel et al., 2016). This leads to an inability to successfully locate similar users and hence leads to the generation of weak recommendations.

In this paper, we present a new collaborative filtering model that does not depend on users' rating. Our novel method computes the similarity between users according to the users' profiles represented as Dynamic Normalized Tree of Concepts (DNTC) model as in our earlier work (Al Alshaikh et al.,

[1] http://www.mendeley.com/
[2] https://www.netflix.com/gb/

Al Alshaikh M., Uchyigit G. and Evans R.
Predicting Future Interests in a Research Paper Recommender System using a Community Centric Tree of Concepts Model.
DOI: 10.5220/0006502900910101
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017)*, pages 91-101
ISBN: 978-989-758-271-4

2017). The concepts are the categories in the 2012 ACM CCS ontology (ACM, 2012). The similarity is computed by using the Tree Edit Distance algorithm (Lakkaraju et al., 2008). Then, a Community-Centric Tree of concepts (CCT) is created. The CCT is used to recommend a set of papers that may relate to the user's future interests. We conducted offline evaluations using the BibSonomy dataset (Knowledge & Data Engineering Group, 2017), which contains actual records of users' posts of research papers. Our model is compared with two baselines: content-based DNTC (Al Alshaikh et al., 2017) and User-based Collaborative filtering (UBCF) as in (Nadee et al., 2013). Our model significantly outperforms the two baselines. This is because it maintains the parent-child relationships between the concepts from the 2012 ACM CCS ontology, it considers other potential interests that can be extracted from similar users to the target user, and it avoids the problem of sparsity. The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 presents our model. Section 4 presents evaluations and results. Finally, the conclusions and future work are presented in section 5.

2 RELATED WORK

Most recommender systems in the research paper domain use content-based approaches; for example, the systems that are developed by Chandrasekaran et al. (2008), Kodakateri et al. (2009), Tang and Zeng (2012), and Al Alshaikh et al. (2017). Each of these approaches use ontologies in their user profiling models. Using ontologies provides a significant improvement in the performance of the recommender systems (Gauch et al., 2007). Gauch et al. (2007) noted that most researchers who used ontologies for user profile representation use them in a similar way to weighted keywords where the concepts are represented as vectors of weighted features. Tang and Zeng (2012) and Kodakateri et al. (2009) use vectors of concepts from a predefined ontology to represent user profiles. The ontology that is used in (Tang and Zeng, 2012) is from Sciencepaper Online (Sciencepaper, 2012). Kodakateri et al. (2009) use the '98 ACM CCS ontology (ACM, 1998). The vector of concepts method assumes that the concepts are independent of each other, which is not an accurate representation of the user's preferences (Chandrasekaran et al., 2008). Chandrasekaran et al., (2008) represents the user profile as a tree of concepts. In this technique, the parent-child relationships between the concepts from '98 ACM CCS ontology are maintained whilst computing the similarity between a user profile and the new research papers to be recommended. However, their user profiling model using the tree of concepts technique is static over time, whereas user preferences and needs are not static but change over time. Moreover, this user profiling technique does not normalize the concept weights. Without normalization, the weights in the user's tree of concepts profile representation are too large to compare accurately with the weights in a tree of concepts for a paper in the recommendation phase. To overcome these problems, Al Alshaikh et al. (2017) developed the Dynamic Normalized Tree of Concepts (DNTC) model for user profiles using the 2012 ACM CCS ontology.

Content-based approaches can capture users' current interests, then recommend a set of papers that may related to their current interests. However, content-based approaches are not able to predict users' future interests. Collaborative filtering approaches have the ability to explore potential future interests. There are two major categories of collaborative filtering approaches: the *memory-based* and *model-based* approaches (Shi et al., 2014; Isinkaye et al., 2015). The *memory-based* approaches involve user-based or item-based techniques. In user-based techniques a user-item rating matrix is given, then a user-based technique predicts a user's rating on a target item by combining the ratings that similar users have previously given to that item (Shi et al., 2014). Item-based filtering techniques predict a user's rating using the similarity between items and not the similarity between users. It builds a model of item similarities based on information about other items that a user has previously rated (Deshpande and Karypis, 2004). *Model-based* approaches use the ratings in user-item matrix as input to train prediction models (Ekstrand et al., 2011). These trained prediction models are used to generate recommendations for the users. For example, the matrix factorization model is used in (Gordon et al., 2008) and feedforward neural network model is used in (Vassiliou et al., 2006).

Nevertheless, the existing collaborative approaches are not appropriate for the research paper domain because they depend on a large number of users' rating, where there is a lack of rating in research paper domain (Yang et al., 2009 and Beel et al., 2016). Nadee et al. (2013) tried to solve the lack of users' rating problem in book recommendation domain. They presented a recommendation approach that considers both the similarity between users and

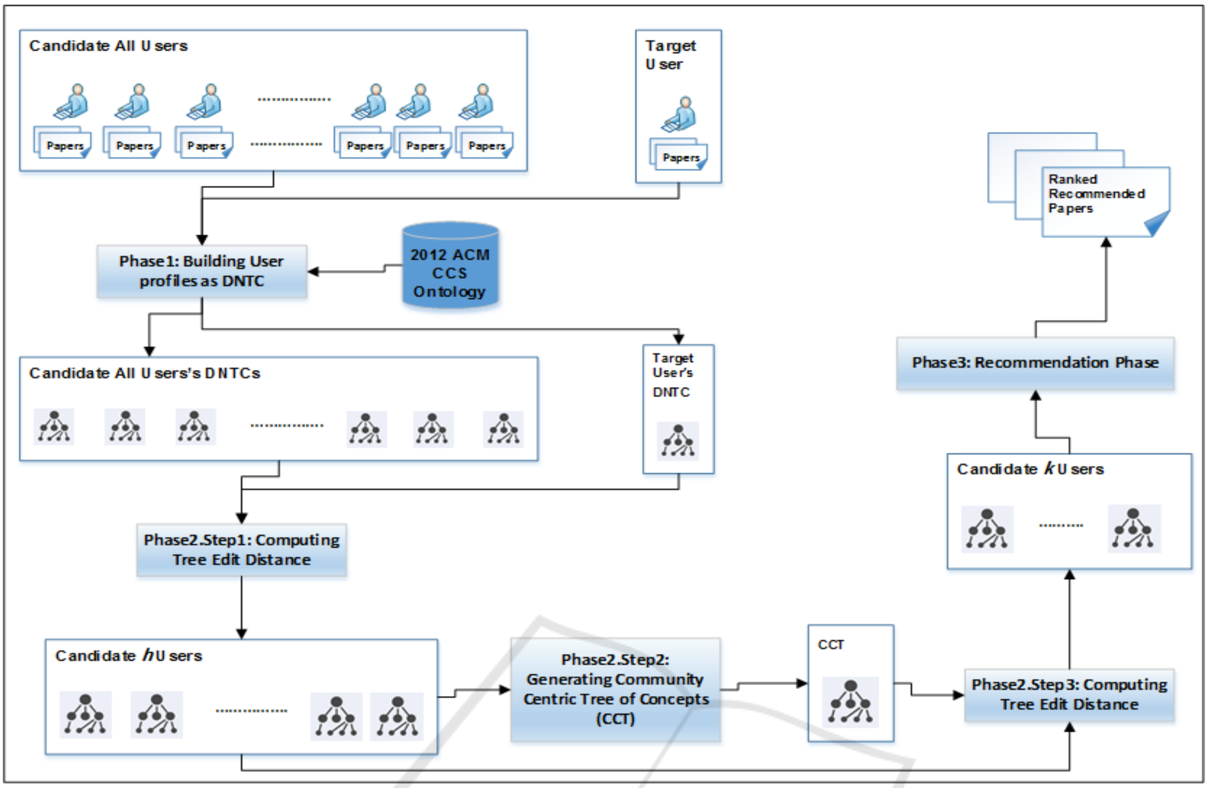
Candidate All Users
Papers
Target User
Papers
Ranked Recommended Papers
Phase1: Building User profiles as DNTC
2012 ACM CCS Ontology
Candidate All Users's DNTCs
Target User's DNTC
Phase3: Recommendation Phase
Candidate k Users
Phase2.Step1: Computing Tree Edit Distance
Candidate h Users
Phase2.Step2: Generating Community Centric Tree of Concepts (CCT)
CCT
Phase2.Step3: Computing Tree Edit Distance

Figure 1: Our collaborative recommendation model.

items, and items' popularity to overcome the overspecialization problem. However, their recommendation results are not sufficiently effective for research paper domain. To overcome the problem of lack of users' rating, we have developed a new collaborative filtering model that does not depend on users' rating, which we introduce in the next section.

3 OUR MODEL

The proposed recommendation model is comprised of three phases:

1- Building user profiles as Dynamic Normalized Trees of Concepts using the 2012 ACM CCS ontology.
2- Computing the similarity between the target user and candidate users, then generating a "Community-Centric Tree of concepts" (CCT) for the target user.
3- Recommending a ranked list of research papers for the target user based on CCT.

Figure 1 presents our collaborative recommendation model.

3.1 Phase 1: Building User Profile as DNTC

The main goal of this phase is to build a user profile as Dynamic Normalized Tree of Concepts (DNTC) as in our earlier work (Al Alshaikh et al., 2017). The BibSonomy dataset is used to create a database of users and the papers which they have read. This phase involves two steps: classifying the papers read by the users to the related concepts in the 2012 ACM CCS ontology and building a DNTC profile for each user.

3.1.1 Classifying Papers

The papers that are read by the users are classified to create profiles of the papers for the recommender system. For classification, we used the TF-IDF weighting algorithm and cosine similarity in our classifier (Al Alshaikh et al., 2017). The cosine similarity (SW_j) between a paper and a concept c_j is the degree of association between the paper and the concept c_j. Each paper in the BibSonomy dataset is classified to the three most closely related concepts in the 2012 ACM CCS ontology and stored in the paper's profile along with their cosine similarity. The resulting profile of each paper is stored in the database which is used to build the DNTC profile for each user.

3.1.2 Building DNTC for Each User

Building a user profile as a DNTC maintains parent-child relationships between the concepts from the ontology. These relationships can be useful while computing the similarity between two users' profiles. For each paper that is read by the user, the top three related concepts and their corresponding cosine similarity weights are retrieved from the paper's profile, which results from the classification phase. In order to exploit the relationships between concepts in a hierarchical concept ontology, a user tree of 2012 ACM CCS ontology is initiated with zero weights for all concepts. Then, the user tree is updated each time a new paper is read by the user as follows. For every new paper, the top three concepts and their corresponding cosine similarity weights (*SW*) are used to update the existing user tree. First, the *SW* weights for the top three concepts are updated by adding the new *SW* weights to old weights values in the user tree. Then, new weight values recursively propagate to the parent nodes until the root node is reached. We assign weights to parents according to the following equation:

$$\mathrm{SW}_{\mathrm{Parent}} = \propto \times\ \mathrm{SW}_{\mathrm{Child}} \qquad (1)$$

Where SW_{Parent} is the weight of the parent, SW_{Child} is the weight of the child and α is the weight propagation factor. α is used to maintain the parent-child relationships between the concepts in the user's tree and its value varies between 0 and 1. Al Alshaikh et al. (2017) found that the best value of α is 0.4. Then, all concept weights are divided by the total number of papers that are read by the user in order to normalize the concept weights. The output of this step is a normalized tree of concepts and its corresponding weights for each user.

3.2 Phase 2: Computing the Similarity between Users and Generating CCT

The purpose of this phase is to determine the community of users whose user profiles are similar to the target user. There are three steps in this phase as follows.

3.2.1 Step 1: Find a Set of *H* Most Similar Users to a Target User

The similarity between a target user and the candidate user is computed using the Tree Edit Distance algorithm (Lakkaraju et al., 2008) to calculate the distance between two DNTC trees (a target user's DNTC and a candidate user's DNTC). This distance is the cost of transforming one tree into another with the minimum number of operations. There are three types of operation: insertion, deletion and substitution. The insertion operation is the cost of inserting a new concept into the tree with a given weight. The deletion operation is the cost of deleting an existing concept with a given weight from the tree. The substitution operation is the cost of changing a concept's weight to another weight. In the 2012 ACM CCS trees we suppose that the concept with zero weight is non-existing node. Hence, the cost of deletion or insertion of a concept is equal to the weight associated with the concept. By contrast, the substitution cost is the difference between weights of an existing concept in both trees. Thus, we calculate the cost of modifying a DNTC tree for a candidate user to match a target user DNTC tree. The two most similar DNTC trees are those which have the lowest total cost of transformations between them. After calculating the total cost between all DNTC trees for candidate users and a target user DNTC tree, the total cost together with its associated id of the user (UserID) are stored as list and these are sorted in increasing order. Hence, the closest candidate user to the target user appears first in the list and the most distant candidate users appear last. Then, the most *h* similar users are selected and stored as set h_i for a target user *i*. *h* is a parameter that will be evaluated in experiments in section 4.2.

3.2.2 Step 2: Generating "Community Centric Tree of Concepts"

The selected *h* similar users are used to generate a Community Centric Tree of Concepts (CCT). The CCT is generated by combining the *h* users DNTC profiles as follows. First, *CCTi* for a target user *i* is initialized as tree of 2012 ACM CCS concepts with zero weights for all concepts. Then, the weights for all concepts from all *h* similar users are summing up. Finally, all concept weights are divided by the number of *h* similar users in order to normalize the concept weights. CCT_i represents the centric of the community interests for the target user *i*.

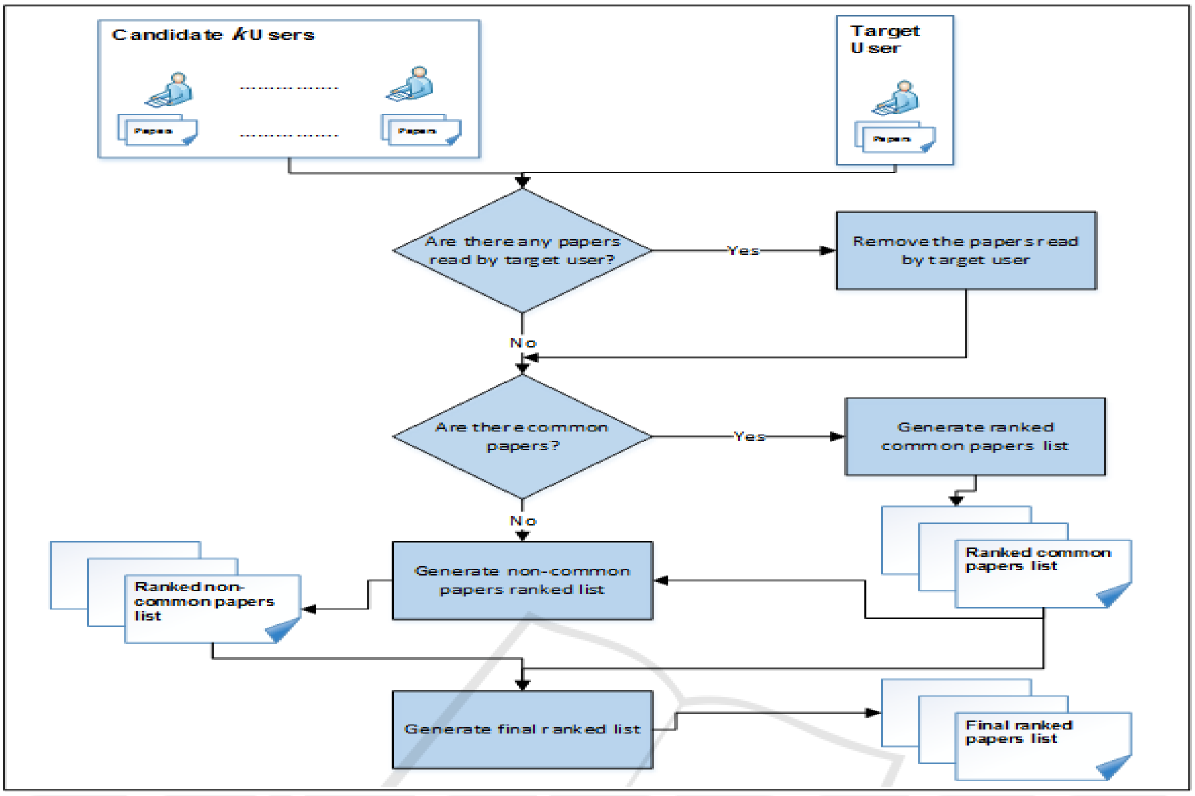
Candidate *k* Users
Papers
Papers
Target User
Papers
Are there any papers read by target user?
Yes
Remove the papers read by target user
No
Are there common papers?
Yes
Generate ranked common papers list
No
Ranked common papers list
Generate non-common papers ranked list
Ranked non-common papers list
Generate final ranked list
Final ranked papers list

Figure 2: Flowchart for recommendation phase.

3.2.3 Step 3: Find the *K* Most Similar Users (from the Set *H* Users)

In this step, we use *CCTi* to find the closest users from the set h_i to the centric of the community interests. The similarity between *CCTi* and the users in the set h_i is computed by using the Tree Edit Distance algorithm. After calculating the total cost between *CCTi* and DNTC trees for the users in the set h_i, the total cost with its associated id of the user (UserID) are stored as a list and sorted in increasing order. Hence, the closest user to *CCTi* appears first and the most distant user appears last. Then, the *k* most similar users are selected and stored as set k_i for a target user *i*. The set k_i is a subset of the set h_i. k_i is a parameter that will be evaluated in experiments in section 4.2. Evaluation results in section 4.2 show that using the set k_i for making recommendations produces better results than using the whole set h_i. This is because the set k_i represents the users that are closer to the CCT_i, which represents the centric of the community interests.

3.3 Phase 3: Recommendation Phase

In this phase, a ranked list of the *top N* research papers is recommended to a target user *i*. First, the papers that are read by the users in the set k_i are retrieved from the database as set Pk_i. If there are any papers already read by a target user *i*, then those papers are removed from the set Pk_i. Then, the set of papers Pk_i is ranked as follows:

a- If some papers appear more than once in the set Pk_i, that means there are common papers between more than one user in the set k_i. The number of appearances of each common paper CP_j in Pk_i is calculated as NCP_j. Then, the papers in Pk_i are ranked according to NCP_j in descending order. Hence, the most common papers have higher ranks. We call this ranked list the common papers list.

b- If there are no common papers (or the common papers are fewer than the number of *top N* recommended papers), then the content-based model is integrated with our collaborative model as follows. We compare the non-common papers profiles with a target user profile. First, a paper profile is represented as tree of concepts as in (Al Alshaikh et al., 2017). Then, the Tree Edit Distance cost is computed between a target user's DNTC tree and the trees of concepts for the non-common papers. We order the papers according to the tree edit distance cost between the paper and the target user's DNTC in increasing order. Hence, the closest papers to a target user appear first and the most distant papers appear last. We call this ranked list the non-common papers list.

The final recommended list that results from the recommendation phase can include both lists: common papers list and non-common papers list. The common papers list appears first before the non-common papers list. Figure 2 shows the flowchart for the recommendation phase.

4 EVALUATION AND RESULTS

In this section, first the evaluation methodology is explained. Then, our model parameters are evaluated to find optimal values. Finally, we compared our proposed model against two baselines

4.1 Evaluation Methodology

We evaluated the performance of our proposed model using the BibSonomy dataset that contains actual records of users' interests as posts for research papers over approximately a ten-year period. Each post contains: metadata for a research paper, date and time of the post. We consider these posts as users' reading records of research papers. We used users' records for the last two years 2015 and 2016 for users in computing area. This includes 1,642 users and 43,140 research papers. Each paper is classified to the three most closely related concepts from the 2012 ACM CCS ontology. A target user's record is divided into a training set of papers (60%) and testing set of papers (40%). The training set are papers that were read by the user before the testing set. The precision for cut-off results at position *N* (P_N) is used to evaluate the *top N* recommended papers. The purpose of our paper is to evaluate the future interests/concepts for a target user. Therefore, our precision metric for the future concepts of interest is defined as follows.

Assume a set $FC = \{FC_1, FC_2, \ldots\ldots, FC_m\}$ is a set of future concepts, *m* is the number of future concepts. A future concept is a concept that does not exist in a target user's training set as shown in Figure 3. The precision for a future concept (FC_i) is defined as follows:

$$P(FC_i)_N = \frac{\text{Number of relevant recommended papers to } FC_i}{N} \qquad (2)$$

Then, the average precision (AP_f) for *m* future concepts for a user is calculated as follows:

$$AP_f = \frac{P(FC_1)_N + P(FC_2)_N + \cdots + P(FC_m)_N}{m} \qquad (3)$$

The mean average precision for all users is calculated as follows:

$$MAP_f = \frac{\sum_{i=1}^{U} AP_{fi}}{U} \qquad (4)$$

where *U* is the total number of users. The top 10 recommended papers are evaluated in our experiments.

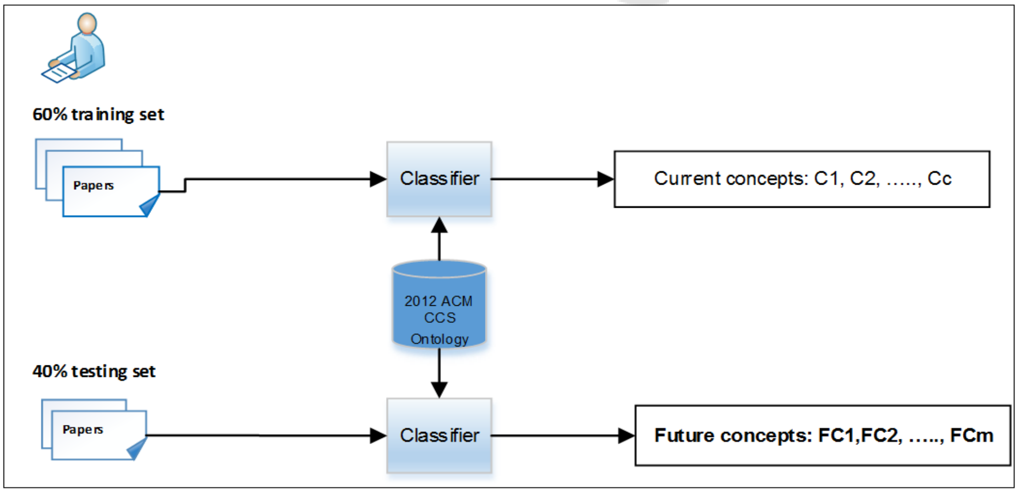

Figure 3: Future concepts.

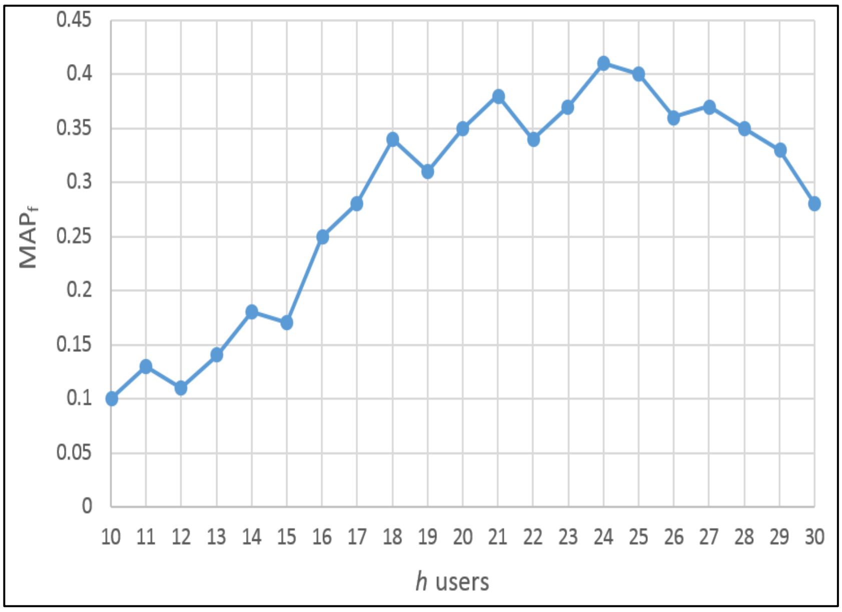

Figure 4: MAP_f results without CCT for different values of *h*.

Precision is an appropriate type of measurement for systems that only aim at providing highly relevant items to users (Agarwal et al., 2005; Hawalah and Fasli, 2015). Whereas recall and F-measure are not the most appropriate types for these systems for the following reasons. The aim of a research paper recommender system is to present a small amount of relevant information from a massive source of information. Therefore, it is more important to return a small number of recommendations that contains relevant items rather than giving the user a large number of recommendations that may contain more relevant recommendations but also requires the user to select through many irrelevant results. The ratio between the number of relevant results returned and the number of true relevant results is defined as recall. Notice it is possible to have very high recall by making a lot of recommendations. In the research paper domain, a user will be more interested in reading papers that really qualify for his/her interests rather than going through a large list of recommended papers and then selecting those which are of interest. Precision more accurately measures a research paper recommender system ability to reach its aim than recall (Agarwal et al., 2005; Hawalah and Fasli, 2015). Therefore, computing the recall and F-measure usually is not important in a research paper recommender system.

4.2 Evaluating Our Model Parameters

We evaluated our model for two options as follows:

Option1: Without Community-Centric Tree of concepts (Without CCT) (i.e. using the set *h* of users for recommendation phase).

Option 2: With Community-Centric Tree of concepts (With CCT) (i.e. using the set *k* of users for recommendation phase).

First, we have to find the optimal value for *h* in option 1, and optimal values for *h* and *k* in option 2.

Figure 4 shows the MAP_f results of applying our recommender system without CCT. Different values for *h* are tested from 10 to 30 users. It can be clearly seen that the MAP_f results for *h* = 10 are relatively low. This shows that using 10 similar users' papers to be included during recommendation phase is not enough. The MAP_f results increase whenever the *h* value increases until *h*=24. When *h*=24, we have the best result of MAP_f with a score of 0.41. This shows that 24 similar users may hold the most essential concepts that are expected to be related to a target user in future.

Figure 5 shows the MAP_f results of applying our recommender system with CCT using different values for *k* and *h*. We tested our system with different values for *h* from 15 to 30 users. It can be clearly seen that the MAP_f results for *h* = 15 are relatively low. This shows that 15 similar users is a very small number of users to generate CCT using them. The MAP_f results increase whenever the *h* value increases until *h*=21. When *h*=21, we have the best results because 21 similar users may hold the most essential interests to generate CCT. When the *h* value

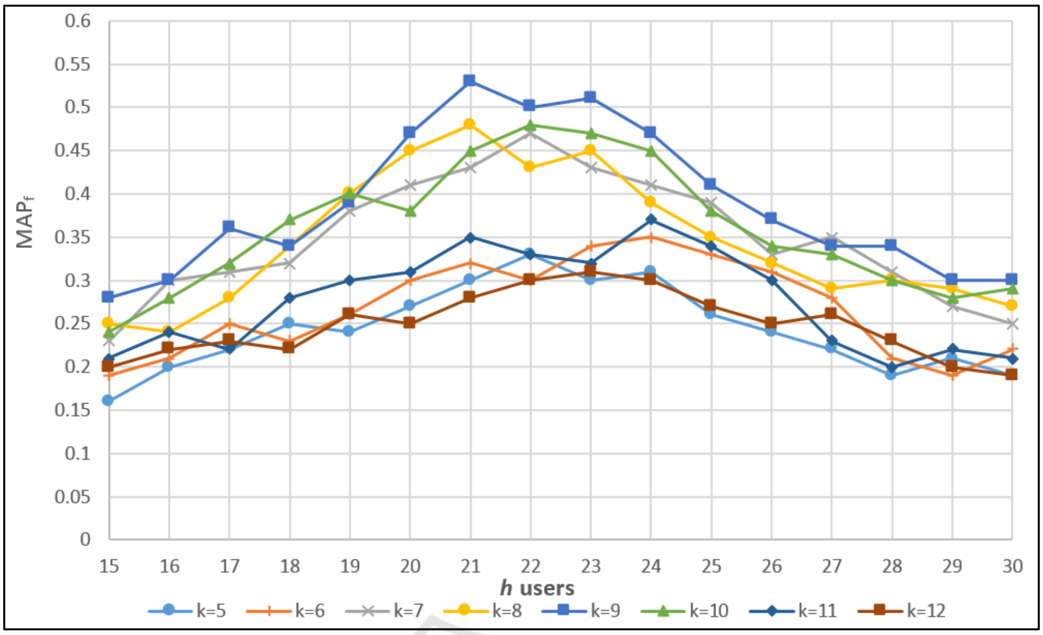

Figure 5: MAP_f results with CCT for different values of h and k.

larger than 21, the MAP_f results tend to decrease, this shows that more than 21 similar users is too large number of users to be included when generating the CCT. We tested our system with different values for k from 5 to 12 users. The MAP_f results improve when the h value comes close to 21 and k values increase.k from 5 to 12 users. The MAP_f results improve when the h value comes close to 21 and k values increase. The results are very low when k = 5, this shows that using only five of the user's papers during recommendation phase is not enough. In general, the best MAP_f results are when k=8, k=9 and k=10. The optimal MAP_f result is 0.53, when h=21 and k=9.

The results show that the best MAP_f value in option 2 with CCT (MAP_f = 0.53) is greater than the best MAP_f value in option 1 without CCT (MAP_f = 0.41). Therefore, using CCT provides better recommendations in our system.

4.3 Evaluating Our Models against Baselines

We compared our proposed model against two baselines

Baseline 1: content-based DNTC (Al Alshaikh et al., 2017): a content based recommender system that compares a user's DNTC profile with unread papers' profiles (which are represented as trees of concepts) to recommend the most relevant papers to the target user's interests. The similarity between a target user and a paper is calculated by Tree Edit Distance algorithm.

Baseline 2: User-based Collaborative filtering (UBCF) as in (Nadee et al., 2013): The user-based collaborative filtering model is based on user-item relationships. The similarity between two users is calculated based on the overlap of their paper sets by using the vector cosine similarity algorithm. The s most similar users are selected. Then, the missing rating for any paper i in target user a is predicted by rating the average from the set of s users' ratings for paper i. The top N papers that have the highest average rating from the set s similar users are selected to recommend to the target user a. To avoid the problem of the lack of user ratings in BibSonomy dataset, we assume that if user a did not read paper i, then the rating $r_{a,i}$ = 0. If user a read paper i, then the rating $r_{a,i}$ =1. The BibSonomy system have an a ttribute that indicate if user a post paper i more than once, hence we assume $r_{a,i}$ = 2, if the user post the paper more than once. We tested different values of s from 10 to 30 users to find the optimal value of s. Figure 6 shows the results for UBCF with different values of s. The best MAP_f is 0.29, when s = 26.

.

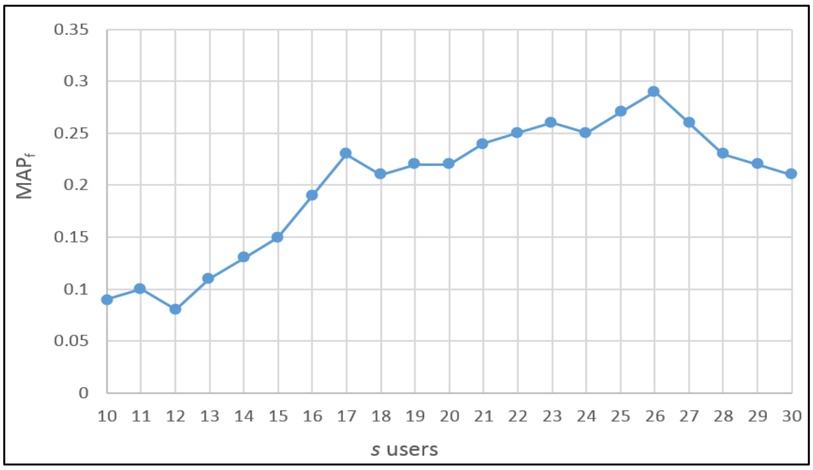

Figure 6: Different values of *s* for UBCF model.

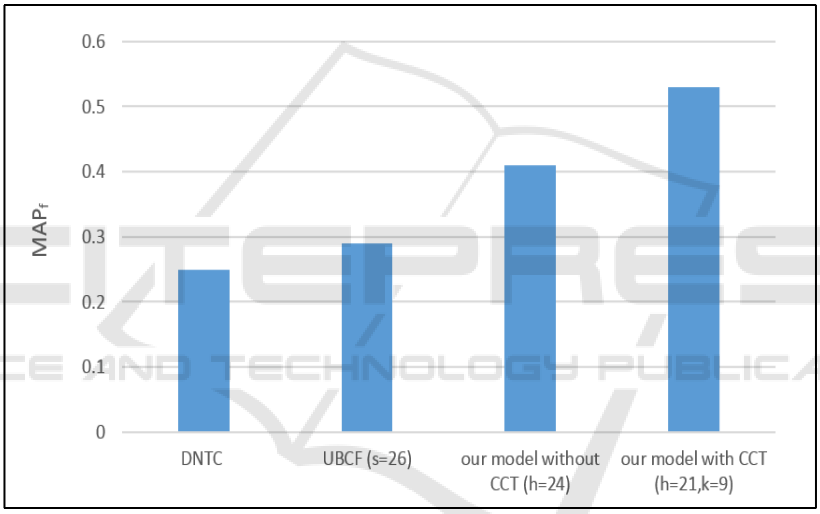

Figure 7: MAP_f results for our model (with and without CCT) against the two baselines.

Figure 7 shows overall comparison results for our system (with and without CCT) against the two baselines. It can be seen that the DNTC model achieves the lowest precision performance with a MAP_f of 0.25. The DNTC model can predict some of user's future concepts because it maintains parent-child relationships between the concepts from the 2012 ACM CCS ontology whilst computing the similarity between a user profile and the new research papers to be recommended. However, DNTC model uses only the current user's interests without considering other potential interests that can be extracted from similar users to the target user.

When it comes to the UBCF model, there is improvement in the performance with MAP_f to 0.29. This model is better than the DNTC model because it considers potential interests that can be concluded from similar users to the target user. However, it has a limitation of sparsity, because UBCF model depends on users rating and the overlap of their paper sets.

Our model (with and without CCT) outperforms the two baselines. This is because it maintains parent-child relationships between the concepts from the 2012 ACM CCS ontology; considers other potential interests that can be extracted from similar users to the target user; and avoids the problem of sparsity. Our model with CCT has better result (i.e. MAP_f = 0.53) than our model without CCT (i.e. MAP_f = 0.41). This is because CCT represents the centric of the community interests.

5 CONCLUSIONS

Current content-based recommender systems suffer from overspecialization problem and they may not have the ability to explore potential future interests. Collaborative filtering approaches can solve this problem; however the existing approaches may not be able to locate successful similar users and result in weak recommendations because of the high sparsity problem in the research paper domain. In this paper, we developed a novel collaborative filtering method that does not depend on users' rating. Our novel method computes the similarity between users according to the users' profiles that are represented as Dynamic Normalized Tree of Concepts using 2012 ACM CCS ontology. Then, a Community Centric Tree of concepts (CCT) is generated and used to recommend a set of papers. We performed offline evaluations using the BibSonomy dataset. Different values for the parameters in our model are tested to find the optimal values. Then our model is compared with two baselines: content-based DNTC and User-based Collaborative filtering (UBCF). Our model (with and without CCT) significantly outperforms the two baselines. Our model with CCT has better result than our model without CCT. In future work, we will improve our model to be hybrid approach by including content-based models that are able to detect short-term and long-term user's interests.

REFERENCES

Agarwal, N., Haque, E., Liu, H. and Parsons, L., 2005. Research paper recommender systems: A subspace clustering approach. *In Advances in Web-Age Information Management*, Springer, pp.475-491.

Al Alshaikh, M., Uchyigit G. and Evans, R, 2017. A Research Paper Recommender System Using a Dynamic Normalized Tree of Concepts Model for User Modelling. *In IEEE Eleventh International Conference on Research Challenges in Information Science (IEEE RCIS 2017).* pp.200-210.

ACM Computing Classification System, 2012, URL: https://www.acm.org/about/class/2012.

ACM Computing Classification System, 1998, URL: http://www.acm.org/about/class/1998.

Beel, J., Gipp, B., Langer, S. and Breitinger, C., 2016. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, pp.305-338.

Chandrasekaran, K., Gauch, S., Lakkaraju, P. and Luong, H.P., 2008. Concept-based document recommendations for citeseer authors. *In International Conference on Adaptive Hypermedia and Web-Based Systems*, pp.83-92. Springer Berlin Heidelberg.

Deshpande, M. and Karypis, G., 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, pp.143-177.

Ekstrand, M.D., Riedl, J.T. and Konstan, J.A., 2011. Collaborative filtering recommender systems. *Foundations and Trends in Human–Computer Interaction*, pp.81-173.

Gauch, S., Speretta, M., Chandramouli, A. and Micarelli, A., 2007. User profiles for personalized information access. *The adaptive web*, pp.54-89.

Gordon, G.J. and Singh, A.P., 2008. Relational learning via collective matrix factorization. *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 650-658.

Hawalah, A. and Fasli, M., 2015. Dynamic user profiles for web personalisation. *Expert Systems with Applications*, 42(5), pp.2547-2569.

Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A., 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, pp.261-273.

Jain, M., 2012. Algorithms for Research Paper Recommendation System. *International Journal of Information Technology*, 5(2), pp.443-445.

Knowledge & Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy, version of January 1st, 2017.

Kotkov, D., Veijalainen, J. and Wang, S., 2016. Challenges of serendipity in recommender systems. *In Proceedings of the 12th International conference on web information systems and technologies.*

Kodakateri Pudhiyaveetil, A., Gauch, S., Luong, H. and Eno, J., 2009. Conceptual recommender system for CiteSeerX. *In Proceedings of the third ACM conference on Recommender systems*, pp. 241-244.

Lakkaraju, P., Gauch, S. and Speretta, M., 2008. Document similarity based on concept tree distance. *In Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pp. 127-132.

Nadee, W., Li, Y. and Xu, Y., 2013. Acquiring user information needs for recommender systems. *In Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pp. 5-8.

Sciencepaper Online, 2012, URL: http://www.paper.edu.cn/en.

Shi, Y., Larson, M. and Hanjalic, A., 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1), pp.3-48.

Tang, X. and Zeng, Q., 2012. Keyword clustering for user interest profiling refinement within paper recommender systems. *Journal of Systems and Software*, pp.87-101.

Vassiliou, C., Stamoulis, D., Martakos, D. and Athanassopoulos, S., 2006. A recommender system framework combining neural networks & collaborative filtering. *In Proceedings of the 5th WSEAS international conference on Instrumentation, measurement, circuits and systems*. pp. 285-290.

Yang, C., Wei, B., Wu, J., Zhang, Y. and Zhang, L., 2009. CARES: a ranking-oriented CADAL recommender system. *In Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pp. 203-212.

Yang, W., Tang, R. and Lu, L., 2016. A fused method for news recommendation. *In IEEE International Conference in Big Data and Smart Computing (BigComp)*, pp. 341-344.