HASHTAG RECOMMENDATION SYSTEM IN A P2P SOCIAL NETWORKING APPLICATION

Keerthi Nelaturu[1], Ying Qiao[1], Iluju Kiringa[1] and TetHin Yeap[1]

[1]School of Electrical Engineering and Computer Science,
University of Ottawa, ON, Canada
{knela006, yqiao074, iluju.kiringa, tyeap}@uottawa.ca

ABSTRACT

In this paper focus is on developing a hashtag recommendation system for an online social network application with a Peer-to-Peer infrastructure motivated by BestPeer++ architecture and BATON overlay structure. A user may invoke a recommendation procedure while writing the content. After being invoked, the recommendation procedure returns a list of candidate hashtags, and the user may select one hashtag from the list and embed it into the content. The proposed approach uses Latent Dirichlet Allocation (LDA) topic model to derive the latent or hidden topics of different content. LDA topic model is a well-developed data mining algorithm and generally effective in analysing text documents with different lengths. The topic model is used to identify the candidate hashtags that are associated with the texts in the published content through their association with the derived hidden topics.

The experiments for evaluating the recommendation approach were fed with the tweets published in Twitter. Hit-rate of recommendation is considered as an evaluation metricfor our experiments. Hit-rate is the percentage of the selected or relevant hashtags contained in candidate hashtags. Our experiment results show that the hit-rate above 50% is observed when we use a method of recommendation approach independently. Also, for the case that both similar user and user preferences are considered at the same time, the hit-rate improved to 87% and 92% for top-5 and top-10 candidate recommendations respectively.

KEYWORDS

Bestpeer, baton, Hashtag, topic model, hit-rate and peer-to-peer networks.

1. INTRODUCTION

Most of the current social networks adopt centralized server architecture. This kind of architecture has both its pros and cons. In centralized architecture, we have all the applications running with their data at one location, at which one or more large computers are connected. Pros include ease of maintenance, any administration or upgrade on the system can be easily done across the components of all the applications. Backup and restore mechanisms are easy to implement since its just one central location and security mechanisms can be incorporated in a simple manner. On the other hand cons include, bottleneck in performance and privacy concerns

David C. Wyld et al. (Eds) : ACITY, DPPR, VLSI, WiMNET, AIAA, CNDC - 2015
pp. 81–93, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51308

from user data perspective. In order to avoid these defects a different line of architecture pattern called the distributed or peer-to-peer architectures are being employed. Peer-to-peer (P2P) systems support for user data privacy, scalability, and availability avoiding single point of failure. Keeping this in view, we are working towards the development of a unique social networking application, which has peer-to-peer architecture. The architecture is inspired from BestPeer++[4] and BATON overlay network [8]. In any social networking application as the user-generated content increases it becomes hard to organize ones own data. Tagging has been a way of organizing data in many of the social networking sites like Facebook[1] and Twitter[2]. We make use of Hashtags, which is one way to tag content. Hashtags are short words with continuous characters without any space in between. They are identified by the presence of '#' before the words. They can be used anywhere within the messages, phrases etc. They have been mainly used for categorizing or highlighting an event, topic, news, individuals etc [15]. This concept has been employed in many social networking sites till date and has become popular with the start of Twitter Social Networking website. Until now these have been used for media broadcasting and business, promotions etc [13]. We developed a hash tag recommendation approach for our online social networking platform to suggest suitable hash tags to a user.

The paper is organized as follows. Section 2 covers the background and related work, which includes details on popular recommendation system techniques.Section 3, illustrates the architecture of our peer-to-peer based social networking application and its high level components. In this section, we discuss the modified implementations of BestPeer++ architecture and BATONoverlay network. Section 4, we discuss in detail the hashtag recommendation methodology.Section 5, presents the details on the datasets used for experiments, the test setup environment and all of the experiments performed to ensure the correctness of algorithms and to calculate the performance of the algorithms. Section 6, we present the conclusion and future work.

2. BACKGROUND AND RELATED WORK

2.1. BestPeer++ Architecture

BestPeer++[4] is a cloud service model. Any business that wants to use the service just has to register themselves and create a BestPeer++ instance, into which they can export data for further processing. This also gives an option for pay-as-you-go query processing model with the help of cloud computing. There are two main components in BestPeer++ - core and adapter. Adapter has two parts, one is an interface to the service and the other part contains adapters, which implement this interface with the help of service provider APIs. The core component consists of query processing and the P2P overlay for serving responses to the queries. There are two kinds of elements in core, bootstrap peer and normal peer. When a business creates an instance, a database server is assigned to that particular instance. This server is then included into the structured P2P overlay arrangement, along with all the other servers. So a normal peer here is a server of a particular business instance. Figure 1 shows the components of the BestPeer++ architecture.

Responsibilities are divided between bootstrap peer and normal peer. The whole network has a single bootstrap peer. This is the server through which normal peers try to join the network. It

[1]Facebook, https://Www.facebook.com/.
[2]*Twitter.* (2015). https://about.twitter.com/company

works like an administrator for the network. Some of the tasks performed by this bootstrap peer are - auto scaling (when an instance exceeds its storage or to perform load balancing), auto fail-over (when a node in the P2P overlay has failed and had to be removed from the network) and the main task of node joining/leaving. For a normal peer, primary effort goes in data loading and indexing. It also does the schema mapping, query processing and execution, along with data loading. When a new business is added to the network, data is loaded from the corporate production to the instance. When this process is being done, normal peer tries to do schema mapping i.e mapping the local business schema to the global peer schema.All the normal peers are organized in P2P overlay called, BATON(Balanced tree overlay network) [8]. This is the crux for BestPeer++ functionality. It provides the interface for node joining, leaving, adding or removing data etc. It arranges all nodes in tree structure. BATON allows for processing both exact and range queries. BATON also provides for three types of indexes - table, column and range. BestPeer++ also provides for role-based distributed access control. When a node fails, all the queries are held up until the backup is restored on to the system. With all these features, cloud computing, database and P2P overlay support BestPeer++ is highlighted as a better data sharing application than any other P2P data sharing systems available. Hence, we choose the same for our P2P social networking application.

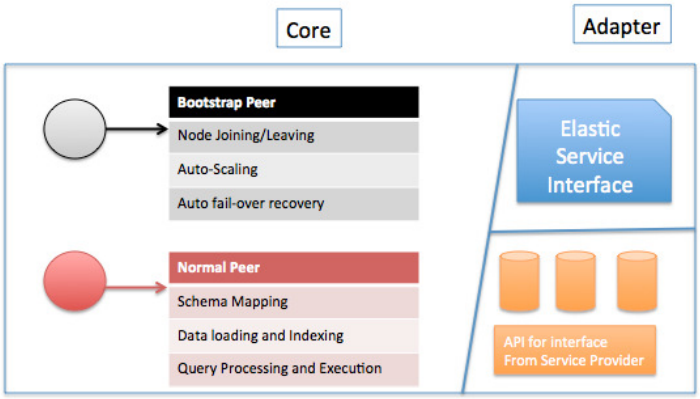

Figure 1: BestPeer++ Architecture components

2.2. Recommendation Systems

Recommendation system is a facility that has been used in web applications for “predicting the user responses to options”[11]. It involves the technique, which is used to make suggestions to the users based on certain selected criterion. Recommendation methods have been classified into two major types: Content-based and Collaborative Filtering. They focus on different perspectives while making recommendations. Content-based methods are specifically on the “properties of items”, and collaborative filtering methods are on the “relationship between users and items”[11]. Content-based recommendation systems (CBR), base their suggestions on the similarity between the items. One major reason behind this content-based approach is that a user always selects similar items. Therefore, during the recommending operation, these systems match the profiles of the items with the profiles of the users. Content-based recommendation becomes better approach for suggestions in a system where there are not many users. However, these systems also have some corresponding limitations [12]. First, the annotations that are added to the content either automatically or manually always will have limited details. It has been seen that the keywords identified for web pages might not contain any information about the media embedded in these web pages. Another limitation is termed as over-specialization. When recommendation is based

on the content already rated by a user, the concentration is restricted to the area already visited by the user. The data outside the domain of the likes of the user might not be considered. Although the dependency with other users in the system is reduced, for a new user, proper recommendations cannot be made until sufficient data about the user's interests have been collected.

Collaborative-filtering recommendation systems, base their suggestions on the similarity of the user's choices on two items. For example in [5], Collaborative-filtering (CF) method employs the nearest-neighbour algorithms to recommend products to a target customer based on the preferences of the neighbours, who have similar interests as of this customer. Though CF methods avoid some of the limitations of the CBR methods mentioned above, there are some drawbacks even with the CF methods. One of them is the same as CBR methods. In order to compare, the interests of a new user with those of others, the CF methods need the information about the ratings or items the user is interested is needed. Other problems are related to the new content added to the application and also sparsity issues. For a new item, it takes some substantial amount of time for the system to collect rating details from other users. Some content might be rated high by a small number of users with peculiar interests. Considering user's profile information apart from the rating data will avoid such scarcity issues.

To avoid limitations of different techniques, many applications implement hybrid recommendation strategies wherein they use both content-based and collaborative filtering techniques are adopted. For our hashtag recommendation, we use hybrid recommendation technique.

2.3. Popular recommendation system techniques

Term Frequency-Inverse Document Frequency (TF-IDF) is the method generally used in text mining and information retrieval systems [3]. It calculates the importance of a word in a document against a corpus. TF-IDF also is offset by the frequency of the word in the corpus. A vector space model indicates the weight in this kind of measure. Term Frequency (TF) of a term in a particular document measures the number of times a term appears. Inverse document frequency (IDF) of a term is calculated upon overall corpus not just one document. It gives the importance of a term in the complete document corpus. Below equation shows the TF-IDF weight calculation, where $m_{ij}$is the number of times a term (t_i) appears in document (d_j), m_i is the number of documents the term has appeared in. M is the total number of documents in the whole corpus[14].

$$tf_{ij} = \frac{m_{ij}}{\sum_k m_{kj}} \qquad idf_i = log\frac{M}{m_i} \qquad TF - IDF_{ij} = tf_{ij}.idf_i$$

In this method, generally document length also plays as a factor. Longer documents tend to have higher values due to the increased number of words and word repetitions. Hence, while calculating the weights of the terms, this approach always normalizes these weights with the length of the documents.

The other technique generally used by recommendation systems is topic models. Topic models are based on the idea that documents are the mixture of topics, where a topic is a probability

distribution over words[17]. A topics model is a generative model. In a generative model, a joint probability distribution is defined over a set of observed and hidden random variables. The joint distribution can be used to generate observable random variables in a generative process. Furthermore, a conditional distribution on hidden random variables can be obtained with the use of the joint distribution and the observed variables. The conditional distribution is also termed as posterior distribution[2]. A topic model always revolves around word and document distributions progressively.

Latent Dirichlet Allocation (LDA) is the one of the simplest topic models. The intuition for LDA is the same as all the other topic models. But the main characteristic is that, all the documents in LDA share the same set of topics. Each document has a probability over each of these topics. The computational problem for LDA is to observe a set of documents and identify the topic-document and topic-word distributions. These probability distributions can further be used for inferring the topic structure of any other documents. LDA also follows the generative model definition. In LDA, the observed variables would be the words of the documents, and the hidden random variables would be the topics.

Here, we describe LDA more formally as defining the topic mixture for each document i.eP(t/d), with a topic mentioned by a distribution over words i.e $P(w_i/t)$ as shown in below equation, where $P(w_i/d)$ is the probability of ith word in a given document d and $t_i$is the topic and $P(t_i = j/d)$ is the probability of identifying a word(w_i) from topic j appearing in document d. $P(w_i/t_i = j)$ is the probability of picking a word from a topic j.

$$P(w_i/d) = \sum_{j=1}^{t} P(w_i/t_i = j)P(t_i = j/d)$$

The topic-document P(t/d) and topic-word P(wi/t) distributions can be estimated by using a corpus of documents[10]. In general convention, θ denotes the topic distributions and φ denotes topic-word distributions. Gibbs sampling algorithm is one of the approaches used for extracting topics from a corpus. It uses an iterative process, which stops until the target distribution is achieved. In an iterative round, each word in the corpus is considered and the estimations for the probability of assigning that word to a topic is done with below equation, conditioned on other word tokens in the same topic. From this conditioned distribution, a topic is sampled and stored as a new topic assignment[17].

$$P(t_i = j|t_{-i}, w_i, d_i, .)\alpha \frac{C_{w,j}^{WT} + \beta}{\sum_{w=1}^{W} C_{wj}^{WT} + W\beta} \frac{C_{d,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d,t}^{DT} + T\alpha}$$

In the equation, C^{WT} maintains count of all topic-word assignments, C^{DT} has the document- topic assignments, t_{-i} represents all topic-term and document-topic assignments except for the current assignment t_i, for word w_i, αand β are the hyper parameters for the Dirichlet priors, works as smoothing factor for the counts. The estimated distributions can be further used in the operations of a recommendation system.

2.4. Related Work

Most of the recommendation proposals ([20], [9], [6], [19]) use only content-based methodology. Using both content-based and collaborative filtering techniques like our proposal Jieying She and Lei Chen's approach [16] results in better hit-rate. Godin et al[6] even though has good hit-rate results, suggests only keywords for the recommendations. The authors do not consider the hash tags already in use and also no collaborative filtering techniques implemented, which reduces the scope of hash tags considered. As per our knowledge, none of the recommendations developed are for Peer-to-Peer network topology. By using the P2P features like scalability and maintenance our approach could achieve a better performance over the other studies.

3. ARCHITECTURE AND COMPONENTS

BestPeer++ is a two-layered architecture. In the current P2P application we have three-tier architecture - bootstrap peer, server peer and client peer as in Figure 2.

3.1 Client Peer

In our application, each user whoever wants to join the network need to use a client side user interface on their PC or mobile device. This user is called the Client Peer. We do not store any data on the client side. All of the data pertaining to a user is stored in the database on the server side. The User Interface helps the user in interacting with the application.

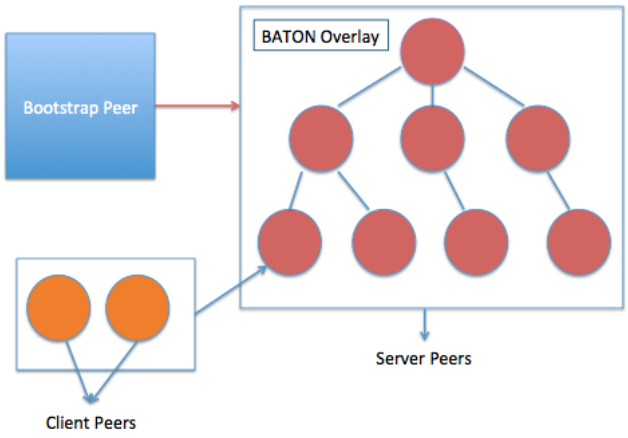

Figure 2: Our P2P Architecture

3.2 Bootstrap Peer

Bootstrap peer in the current architecture, has the same administrator role as in BestPeer++. Single bootstrap peer node accounts for the health of the whole network. Monitors the node joining and leaving. The auto-failover and auto-scaling supposed to be in the bootstrap peer have not been implemented in our system. Users need to register and login via the bootstrap peer each time they connect to the network. Apart from the components in BestPeer++, we also store the user profile and friendships information in Bootstrap peer.

3.3 Server Peer

Client data is stored in the Server Peer nodes. Each server peer is responsible for more than one client node at a time. All of the server peer nodes form the BATON overlay structure. User login information is stored even in the server peers. Major concerns for the server peers lies in the management of P2P overlay structure and user data. Each server is responsible for clients within a particular URI (User Resource Index) range.

Server Peer is the alias for Normal Peer in BestPeer++. Most of the functionalities in normal peer have been imported to server peers with some updates. The schema-mapping module was discontinued as all the data exchanged in the application has the same mapping. Data loader is used during the data retrieval process. Data Indexer is major for the BATON overlay network, as each of the data stored in server depends on the range. It also helps during the forward and lookup requests. For the query execution, we used JPA instead of pure SQL language [18]. BATON tree node information is also stored with the server along with the physical details of bootstrap peer.

4. HASHTAG RECOMMENDATION APPROACH

4.1. Proposed Approach

The proposed hashtag recommendation approach lists out hashtag candidates for a content entered by the user. If no related hashtags are found, this approach may suggest the user with the hashtags that have been used previously or with those related to the user or to the content. The approach also advises the user with some keywords for creating a new hashtag. We adopt a hybrid recommendation system for our social network platform considering both types of recommendation: content and collaborative filtering approaches. Most of the hashtag recommendation systems have lagged in two issues. First, they use only one of the recommendation approaches. In the case that an approach is chosen, a major part of hashtags the user might be interested in is being omitted. For example, the content-based techniques might not include some of the tags being created by similar users or the friends in the suggested tags. Similarly, the collaborative filtering based techniques might neglect the tags related to the posted content or those popular in the overall system. Second, as per our knowledge, none of the ideas reviewed till now have given a user an opportunity to choose the recommendation method he might be interested in.

Hence, considering these drawbacks in the previous research, our approach contains several recommendation modes. The users may control the recommendation system by selecting one or multiple modes. They receive the candidate hashtags recommended by the selected modes. These modes are classified into the following categories. The categories considered are:

1. Global content common for all of the users
2. User preferences evaluated based on their content previously added
3. Hashtags created by users with similar preferences as current user
4. Hashtags created by the friends of the user and are related to the users content being created
5. Overall popular hashtags in the whole social network platform

Also, in the case that any of the methods returns zero tags, the proposed approach even recommends with keywords based on the chosen mode.

4.2. Implementation

Unlike twitter, which has restricted the lengths of the text for its tweets, content with different types in our application can have varied lengths without any restrictions. In such cases, topic model for recommendation systems is a better technique. For our approach we considered to adopt topic analysis technique to evaluate the content similarities and user preferences on content. We further chose Latent Dirichlet Allocation (LDA) process using Gibbs Sampling method for topicclassification. Most of the research in topic models considers only topic-word distributions. The proposed approach goes further to the next step and extracts "topic-hashtag" probability distributions. The LDA process is done in three phases as shown in Figure 3.

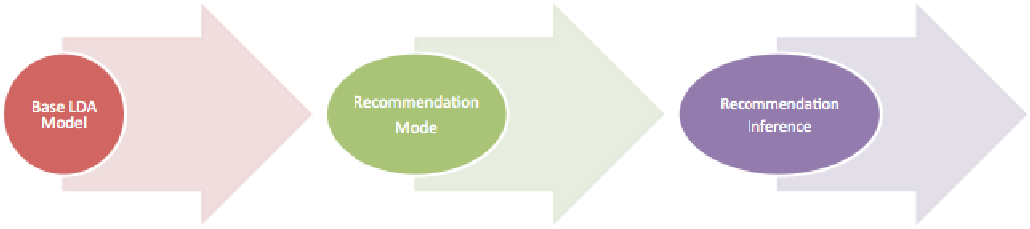

Figure 3: Steps in our LDA processing

Training or Base LDA Model: In the training phase, random content collected across various networks is passed to the procedure implementing the Gibbs algorithm. The procedure estimates the initial topic-word and document distributions of the Base LDA model. The documents in this situation refer to any single post, comment or article.

Estimation or Recommendation Mode: This is phase at which content passed for LDA procedure differs from each of the categories mentioned before. For each mode selected, we pass in the content and generate the updated topic- word, document distributions along with topic-hashtag distribution. For each of the topics and documents, we calculate hashtag distribution over documents can be calculated using below equation where d is the document, h is the hashtag and t is the topic.

$$P(h/d) = \sum_{i=1}^{n} P(h/t_i)\, P(t_i/d)$$

Recommendation Inference: This is the phase in which a user is suggested with candidate hashtags. Content a user enters is passed to the model to evaluate the topic-hashtag distributions and order them according to their probability scores. This procedure also uses the equation above only d refers to the union of the words in the content.

5. EXPERIMENTS AND RESULTS

In this section, we discuss the experiments performed for verifying the correctness of the hashtag algorithms and evaluate the effectiveness of these algorithms.

5.1. Experimental Setup

All of the experiments were performed on a standalone computer with 16GB RAM and MAC OS X or Windows 7 operating systems. We developed a prototype application with JUnit test cases for hashtag implementation. In our prototype, we have two server peers, one bootstrap peer and fifteen client peers. We use JGibbLDA library[3] for performing LDA operations. We set the default values for hyper-parameters $\alpha = 50.0/k$ and $\beta = 0.1$ as suggested in [7] by Griffiths and Steyvers where k is the number of topics considered. For all the experiments we run the LDA operations through 500 iterations of Gibbs Sampling. The contents used by these experiments are, datasets obtained from three different sources. The first was from Textual Retrieval Conference (TREC) 2011 micro blog track[4]. We choose Tweets2011 corpus. This corpus comprises of 16 million tweets collected over a period of two weeks between 24th January 2011 until 8th February 2011. We used Twitter tools API provided by TREC Microblog track to extract tweets. The second source was Twitter web site. We use the Twitter Streaming API to extract tweets. We captured 10000 tweets with trending topics in specific intervals of time for two days. Third source is from Sentiment 140[5]project created by the students from Stanford University for the purpose of Sentiment analysis of topics in tweets. This collection consisted of two datasets: one was training data with 1,60,0000 tweets and one was test data with 500 tweets. Sentiment140 data is pre-processed, where any special characters or emoticons are removed. Before passing the data to the LDA functions, we selected the tweets in these datasets, removed the special characters or any characters other than English letters. The special characters "#" are kept, since it indicating the beginning of a hashtag.

5.2. Experiments and Results

We perform experiments on each of the recommendation modes mentioned in our proposed approach. For evaluating the effectiveness of this recommendation approach, we consider hit-rate of the results from an execution of a recommendation activity as the criteria. A recommendation activity starts from invoking the recommendation function on content upon a user's request to returning the results to the user. The equation to calculate the Hit-rate of the results is defined below. We identify a result as hit if atleast one of the recommended hashtags is a hashtag used for the content.

$$Hit-rate = \frac{Number\ of\ hits}{Number\ of\ content\ items\ considered}$$

There were three sets of experiments performed. For all of the experiments, apart from comparing the actual hashtags used in the content, we also performed subjective evaluation with the five evaluators. The evaluators where asked to mark the recommended hashtags as relevant and non-relevant. Majority views of the votes were considered for the final results.

First experiment compares the hit rate percentage over the number of topics for each of the recommendation modes except for the mode in which we recommend hashtags based on their overall popularity. Figure 4 shows the graph plotted for four categories of recommendations with hit-rate against topics. Initially for all the methods we started of with 50 topics for the LDA.

[3]JGibbLDA, http://Jgibblda.sourceforge.net/.
[4]TREC Twitter2011 datasets, http://Trec.nist.gov/data/tweets/.
[5]Sentiment 140, http://Www.sentiment140.com/.

Since topic-hashtag distribution is the main case that we consider for our proposal, with more topics we expected more hashtags.

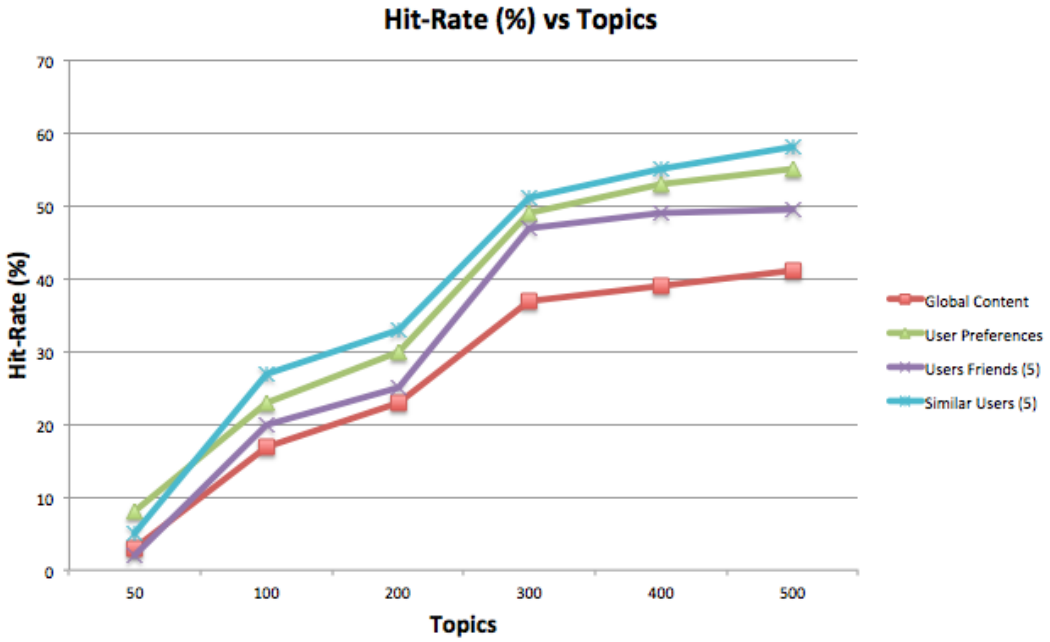

Figure 4: Hit-rate Vs Number of Topics

Recommendation with global content was not satisfactory, as it wouldn't consider any of the user content or user topics of interest. Maximum was 41.3% of hit-rate with global content that too at 500 topics. The results for User Preference based and similar users based recommendations were promising and we were able to see 55% and 57.6% of hit-rate respectively. For the recommendations from similar user and friends we used 5 clients as the users under comparison. Recommendation mode using friends content and interest could give approximately 50% of hit-rate. The other observation we made was on the cases with the number of topics between 300-500 topics; however there was not much of improvement with the results. So, for our application 300 topics would be the ideal number of topics to be considered. Overall Popularity based recommendation needs more server nodes to be evaluated. We were able to set-up only two server nodes. With two server nodes, the algorithm correctness was tested and we were able to retrieve the trending hashtags from the two server nodes.

The second experiment was performed to test the recommendation mode with similar users. This was done using the dataset obtained from Twitter with its Streaming API. As mentioned before we collected 10000 tweets of trending topics from specifically the ones with some of the politicians tagged in them. We wanted to check the maximum number of hashtags out of the total recommendations that would be relevant for the given content. We distributed around 600 tweets to each of the clients, increasing the number of clients at each step. When there was only one user we were not able to retrieve any related recommendations. At 10 and 15 client count we were able to retrieve 4 relevant hashtags of the recommendations made. Hence, as the users increase we would be able to provide with top-k recommendations with k = 5. Figure 5 shows the results for this experiment.

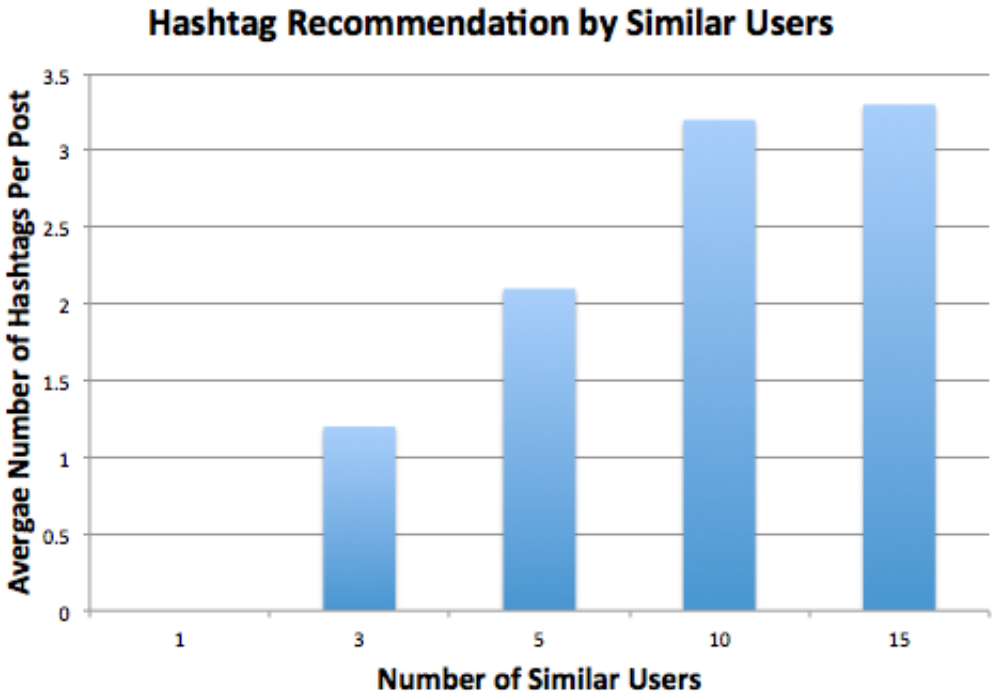

Figure 5: Average hashtags per post Vs Number of similar users

The last experiment was to check hit-rate when we used both user preferences and similar users recommendations at the same time. We used the same data from Twitter Streaming API for this experiment too. We tested for top-k recommendations when k = 5 and k = 10. For hit-1 this choice was good and we were able to acquire around 87% and 92% hit-rate for top-5 and top-10 recommendations respectively. The hit-all rate which checks for a match for all of the hashtags in a tweet, was around 63% when we used top-5 recommendations and it was still below 50% when top-10 recommendations were considered. From the results of this experiment we intuit that may be by combining more than one mode together the proposed approach could provide better results. Table 1 shows the results for the same.

k	Measurement	Values
5	Hit-1	0.87
5	Hit-all	0.63
10	Hit-1	0.92
10	Hit-all	0.49

Table 1: Hit-rate for top-k recommendations with User Preferences and Similar Users

6. CONCLUSION AND FUTURE WORK

In this paper we introduce ourpeer-to-peer social networking architecture and its components. We also propose hashtag recommendation approach proposed for this application using Latent Dirichlet Allocation [3] topic model. It is model, which identifies hidden topics from a set of pre-

processed documents. We specifically concentrate on identifying topic-hashtag distributions out of these hidden topics. These are further used for the recommendations. Our research uses both content-based and collaborative filtering methods for the recommendations, which can be selected by the user on his own choice. Also, we provide the recommendations by considering content from the neighbouring nodes in the network, which would allow us for the fast processing of the recommendations. The experiment results show more than 50% hit-rate for three of the collaborative filtering approaches. The hit-1 rate for top-5 and top-10 recommendations for hashtags considered from similar users and user content is the better than any of the topic model based hashtag recommendation systems. Also, using only similar users method guarantees that the approach is good for top-3 recommendations.

There are some limitations as to the proposed recommendation methodology. We still have to test the performance of the algorithms in peer-to-peer simulated environment with more number of server nodes. Without which we were not able to test the overall popularity method. The next thing would be to consider a top-k recommendation system for all of the methods mentioned. As a future work, we would give the recommendation methods to the user as part of advanced settings and include more than one method for a recommendation. We use relational database for storing both the bootstrap and server peer data. With the users increasing, at some point we need to consider moving to BigData solutions. Also, we need add in encryption mechanisms for securing the client data stored on the server.

ACKNOWLEDGEMENTS

We would like to thank the development team and University of Ottawa for the tremendous support in this research.

REFERENCES

[1] Adomavicius, G., &Tuzhilin, A.(2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.Knowledge and Data Engineering, IEEE Transactions On, 17(6), 734-749.

[2] Blei, D.M. (2012).Probabilistic topic models.Communications of the ACM, 55(4), 77-84.

[3] Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003).Latent dirichlet allocation.J.Mach.Learn.Res., 3, 993-1022.

[4] Chen, G., Hu, T., Jiang, D., Lu, P., Tan, K., Vo, H. T., et al. (2012). BestPeer++: A peer-to-peer based large-scale data processing platform. Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, pp. 582-593.

[5] Chuang, H., Wang, L., & Pan, C.(2008).A study on the comparison between content-based and preference-based recommendation systems.Semantics, Knowledge and Grid, 2008.SKG '08. Fourth International Conference On, pp. 477-480.

[6] Godin, F., Slavkovikj, V., De Neve, W., Schrauwen, B., & Van de Walle, R. (2013). Using topic models for twitter hashtag recommendation. Proceedings of the 22nd International Conference on World Wide Web Companion, pp. 593-596.

[7] Griffiths, T.L., &Steyvers, M.(2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101(suppl 1), 5228-5235.

[8] Jagadish, H.V., Ooi, B.C., & Vu, Q.H. (2005). BATON: A balanced tree structure for peer-to-peer networks. Proceedings of the 31st International Conference on very Large Data Bases, Trondheim, Norway. pp. 661-672.

[9] Jeon, M., Jun, S., & Hwang, E. (2014).Hashtag recommendation based on user tweet and hashtag classification on twitter.Web-age information management (pp. 325-336) Springer.

[10] Krestel, R., Fankhauser, P., &Nejdl, W. (2009).Latent dirichlet allocation for tag recommendation.Proceedings of the Third ACM Conference on Recommender Systems, New York, New York, USA.pp. 61-68.
[11] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014).Mining of massive datasets Cambridge University Press.
[12] Lops, P., De Gemmis, M., &Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender systems handbook (pp. 73-105) Springer.
[13] McFedries, P. (2013). Hashtag, you're it [technically speaking].Spectrum, IEEE, 50(4), 24-24.
[14] Peng, X., Cao, Y., &Niu, Z.2008).Mining web access log for the personalization recommendation.MultiMedia and Information Technology, 2008.MMIT '08. International Conference On, pp. 172-175.
[15] Potts, L., Seitzinger, J., Jones, D., & Harrison, A. (2011). Tweeting disaster: Hashtag constructions and collisions. Proceedings of the 29th ACM International Conference on Design of Communication, Pisa, Italy.pp. 235-240.
[16] She, J., & Chen, L. (2014). TOMOHA: TOpic model-based HAshtag recommendation on twitter. Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion, Seoul, Korea.pp. 371-372.
[17] Steyvers, M., & Griffiths, T.(2007).Probabilistic topic models.Handbook of Latent Semantic Analysis, 427(7), 424-440.
[18] Wang, S. (2014).uopStore: An ecommerce platform with a peer-to-peer infrastructure. School of Electrical Engineering and Computer Science, University of Ottawa).
[19] Yu, J., &Shen, Y. (2014).Evolutionary personalized hashtag recommendation.Web-age information management (pp. 34-37) Springer.
[20] Zangerle E., G. W. Recommending hashtags in twitter.Workshop on Semantic Adaptive Social Web 2011, in Connection with the 19th International Conference on User Modeling, Adaptation and Personalization (2011).