# An Efficient Agent-Based System to Extract Interests of User Groups

Bilal Hawashin and Ayman Mansour

***Abstract*— This work proposes an agent-based system to extract the interests of user groups based on certain category values such as males, teens, singles, and so on. Many previous works proposed recommender systems but their concentration was on users as individuals. Finding interests per group would have two main benefits. First, it would improve the recommender system accuracy. Second, it would give important and interesting information to optimize the market according to the interests of the various groups. Experiments show that the system is efficient in term of the execution time and accuracy.**

***Index Terms*—Agent System, Content Based Filtering, Group Interests, Recommender Systems.**

## I. INTRODUCTION

Recommender Systems are used to suggest items to users based on their interests. They have been used widely in various domains. Examples of recommender systems include research papers recommenders, book recommenders, product recommenders, and many more. In this paper, our concentration is on the domain of online movie stores, even though the proposed system can be used in other domains as well.

Although many works in the literature proposed recommender systems, the majority concentrated on each user as a standalone entity. However, very interesting information can be obtained when we study the behavior and interests of group of users. Users can be grouped according to certain categories. Each category can have certain values. For example, users can be grouped according the Gender category as Males and Females. The interests of each category value can be different from those of other category values. The interests of Males can be different from those of Females. Similarly, users can be grouped according to their Marital Status, Age, Country, Job, and so on. Each of the previous categories can have two or more category values. Both categories and category values are domain dependant. For example, in online movie stores, the age category can have the values Teen, Adult, and Senior.

However, other domains such as clothes stores would include a child value in this category as well. Our method aims at extracting useful information per category value. These information include the most ordered items per category value, the most interesting items per category value, the uniquely ordered items per category value, the trending items per category value, and the terms that represent the interests per category value. For example, what are the most ordered items by teens, what are the most liked items by females, what are the uniquely ordered items in July, what are the trending items by seniors, what are the terms that describe what people in India like. Obviously, this would have a great benefit on the market. Finding the interests of category values would allow the market to optimize itself according to the various needs of category values. Moreover, extracting these interests would contribute in improving the recommender systems themselves. For example, the recommender system would use the feedback from this system to give recommendations based on the time of the year, the region, the user type, and so on.

To improve the system efficiency, we use agent based system. The use of agents would improve the execution time and provide a better modular system. Each agent will be responsible for one category value. A coordinator agent is needed to forward transactions of users to the suitable value agents.

To evaluate the system, and as MovieLens Dataset does not include explicitly the interests of user groups, we used a synthesized dataset of 1000 users and 10000 items. The contributions of this work are as follows.

- Giving interesting information about the interests of each category-value.
- Enhancing the performance of recommender systems.

The rest of this paper is organized as follows. Section II is a literature review of the related works in this field. Section III describes the proposed system. Section IV represents an extension to the system that includes extractors to unique items and trending items per category value. Section V is the experimental part and discussion and Section VI is the conclusion.

## II. LITERATURE REVIEW

Many recommender systems have been proposed in the literature. Content based filters[1],[4],[5],[6],[7] recommend items based on their content similarity to the previously highly rated items by the user. In details, [7] used a threshold to decide whether the description matches that of the highly rated items or not, whereas [5] used the winnow algorithm that works well when many features exist. [4],[6] used Bayesian classifiers to estimate the probability that a user likes an item based on its content. Collaborative filtering [2],[8],[9],[10],[11],[12],[13] on the other hand, uses user user similarity and suggests items that were highly rated by similar users. In details, Tapestry system[9] demanded users to specify their similar users manually. The Grundy system [8] proposed the use of user stereotypes. Memory based methods[10],[11] in collaborative filtering

Manuscript received July 18, 2016; revised Augest 01, 2016. This work was supported by Alzaytoonah University of Jordan.

Bilal Hawashin is with the Department of Computer Information Systems, Alzaytoonah University of Jordan, Amman, 11733 Jordan, email: b.hawashin@zuj.edu.jo.

Ayman Mansour is with the Department of Communication and Computer Engineering, Tafila Technical University, Tafila, 66110, Jordan, e-mail: mansour@ttu.edu.jo.

use the previously rated to find similar users, while model based algorithms[12],[13] learn a model from the previous rates, such as Baysian model[12] and maximum entropy model[13]. Hyprid methods[14],[15] combine both content and collaborative features together. Context aware recommender systems [16],[17],[18] are those that consider context information such as location [17], time[18], and user interests[19] in their recommendations.

As for interest-based recommender systems, our work in [30] extracted user interests as explicit terms to improve recommender systems. [21] developed a reinforcement learning strategy for market based multi agent recommendation system when many recommender systems are used. [20] proposed a design framework for multi agent interest based system. [22] used user movie genre interest to detect account hacks, as attacker would give random and different genre interests .

As for trending topics, they have been studied widely in the literature. Mainly, they have been used with social network mining. They can be divided into two categories: Document pivot methods and Feature pivot methods. Document pivot method produce clustered set of documents. Examples of such works include [23],[24],[25]. On the other hand, feature pivot methods extract the most important terms that represent the text. Examples include [26], which used Latent Dirichlet Allocation (LDA), [27] which proposed extensions to LDA, [28] which used keyword burstiness, and [29] which applied df-idf for each term in each time interval and used wavelet detection methods. Up to our knowledge, no work used the trending topics concept in the improvement of recommender systems.

## III. THE PROPOSED GROUP INTEREST EXTRACTOR

The proposed system aims at extracting the following types of information:

- The most ordered items per category value.
- The most liked items per category value.
- The most liked terms per category value.
- The uniquely ordered terms per category value.
- The uniquely liked terms per category value.
- The trending items per category value.

First, both the categories and category values are defined according to the domain. Next, when a user creates an account, (s)he will be asked to give some personal information related to these predefined categories. For example, if the categories are gender, age, and marital status, the user must select the value that best represents him/her from the list of values in each category. The user information would be stored in a central agent named the coordinator. Besides, each value in each category would have one specific agent responsible for it. Each value agent has an array named Likes of size I, where I is the number of items. Also, it has an array named Orders of size I as well. They are both initialized with zeros.

Next, when a user orders or rates an item, a record containing the user information, the item information, and the rate (if it is a rate operation) is transferred into the coordinator. The coordinator would lookup user information and forwards accordingly the record into its corresponding agents. For example, if the user is a single teen, the coordinator would send the record to the agent responsible for the value single and the agent responsible for the value teen. The agent responsible for the value adult ,for example, would not be concerned in this transaction, and therefore, the record would not be sent to it.

Next, each of the corresponding agents would change its variables according to the record they have received. If the record is an order record, the agent would increment the corresponding item index in the array *Orders* as follows.

$$orders[i] = orders[i] + 1, \quad (1)$$

where *i* is the item number.

On the other hand, if the record is a rate record, the agent would update the *Likes* array by adding the user rate to the corresponding item index as shown in the following formula.

$$Likes[i] = Likes[i] + rate, \quad (2)$$

where i is the item number and rate is the user rate for that item.

Besides, if the record has a rate, the agent would converts the rate into a rate sign. The rate sign indicates whether the user found the item interesting or not. For this purpose, we adopt the use of the mean to indicate the level of interest. If the rate of the user is above three(out of five), this indicates that the user found the item interesting, and the rate sign would be +1. In contrast, if the rate is below three, this indicates that the user found the item not interesting, and the rate sign would be -1. and each agent would store the vector of terms representing this item. The result would be a matrix, *Transaction Tern Matrix*, where every row represents a transaction and every column represents a term. Each record has also the rate sign at the end. Table III is an example, assuming that *Item Term Matrix* is in Table I and *User Transactions* are in Table II.

TABLE I.
*ITEM TERM MATRIX*, WHERE EVERY ROW REPRESENTS AN ITEM AND EVERY COLUMN REPRESENTS A TERM. THE SET OF ALL TERMS ARE EXTRACTED FROM THE ITEM DESCRIPTIONS.

| | | **Terms** | | | | |
|---|---|---|---|---|---|---|
| | | **Child** | **War** | **Tom Hanks** | **James Cameron** | **…** |
| **Items** | **Saving Private Rayan** | 0 | 1 | 1 | 0 | |
| | **Home Alone** | 1 | 0 | 0 | 0 | |
| | **Avatar** | 1 | 0 | 0 | 1 | |
| | **…** | | | | | |

TABLE II.
TRANSACTIONS OF USERS

| **Transaction** | **User** | **Item** | **Rate** |
|---|---|---|---|
| **Tr1** | User1 | Saving Private Rayan | 4 |
| **Tr2** | User2 | Home Alone | 1 |
| **Tr3** | User1 | Avatar | 5 |
| **…** | | | |

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE III.
THE RESULTING *TRANSACTION TERM MATRIX*.

| Trans. | Child | War | TomHanks | JamesCameron | Rate Sign |
|---|---|---|---|---|---|
| **Tr1** | 0 | 1 | 1 | 0 | +1 |
| **Tr2** | 1 | 0 | 0 | 0 | -1 |
| **Tr3** | 1 | 0 | 0 | 1 | +1 |
| ... | | | | | |

Periodically, for each value agent, to extract the most ordered item(s) per a category value, the items with the maximum values in the *orders* array can be extracted. Similarly, the most liked items per category value can be extracted using the *likes* array. In order to find the terms that express the interests of a category value, the user interest extractor, presented in our previous work[30], can be applied to the *Transaction Term Matrix* to get the interests as terms.

The following is an example that illustrates the operations done when a user rates an item.

**Example1.**

A User x(Single Male Teen ) rated Ice Age 4/5.
**Step1:** The coordinator receives User, Item, and Rate
**Step2:** The coordinator forwards Item and Rate to all related value agents(Single Agent, Male Agent, and Teen Agent).
**Step3:** Each of the three agents apply likes[Ice Age] = likes[Ice Age]+4
**Step4:** Each of the three value agents insert a record having Item term vector along with rate sign +1 to matrix *Transaction Term Matrix*

## IV. EXTENDING THE SYSTEM TO ADD THE TRENDING AND UNIQUE ITEMS PER CATEGORY VALUE

The following subsections illustrate the extraction of unique items and trending items per category value.

### *A) Unique Items Per Category Value Extractor*

Unique items per category value can be divided into two parts; uniquely ordered items per category value and uniquely liked items per category value.

*Uniquely Ordered Items Per Category Value*

These are the items that were uniquely ordered by a certain category value. In order to extract them, a matrix need to be created where every record represents the transactions of a person and every column represents an item. If the person ordered an item, 1 will be placed in the person item cell intersection, 0 will be placed otherwise. Attached to each record is the category value indication, +1 if the person is a target person and -1 otherwise. For example, to extract unique items ordered by teens, each person(teen or non teen) would have a record presenting the items he ordered, and at the end of the record, +1 if the person is teen and -1 otherwise. Table IV is an example. Finally, CHI square can be used to extract the items related to teens. CHI square feature selection is used to extract the items related to the target group. CHI square has been used widely in supervised feature selection, and its equation is the following.

$$CHI(\mathrm{t}) = \sqrt{\frac{(n_{pt+}+n_{pt-}+n_{nt+}+n_{nt-})(n_{pt+}n_{nt-}-n_{pt-}n_{nt+})^2}{(n_{pt+}+n_{pt-})(n_{nt+}+n_{nt-})(n_{pt+}+n_{nt+})(n_{pt-}+n_{nt-})}} \quad (3)$$

Where npt+ and nnt+ are the number of items ordered by the target users and non target users respectively; npt- and nnt- are the number of items not ordered by target and non target users respectively. The items with the highest CHI values are extracted and considered uniquely correlated to that target category value.

TABLE IV.
USER ITEM MATRIX. IF THE USER ALREADY ORDERED THAT ITEM, 1 IS PLACED, 0 OTHERWISE. IF THE USER IS A TARGET USER, THE SIGN WOULD BE +1. OTHERWISE, IT WOULD BE -1.

| User | Avatar | Saving Private Rayan | Ice Age | Up | ... | Teen or non teen |
|---|---|---|---|---|---|---|
| User1 | 1 | 0 | 1 | 1 | | +1 |
| User2 | 0 | 1 | 0 | 0 | | -1 |
| User3 | 1 | 0 | 1 | 1 | | +1 |
| ... | | | | | | |

*Uniquely Liked Items Per Category Value*

As for the uniquely liked items per category value, the same method is used with one exception. When the user highly rates an item, 1 will be placed in the user item intersection. 0 will be placed otherwise.

### *B) Trending Items Per Category Value Extractor*

The Trending topics concept has been used widely in social networks. An obvious example is trending topics in twitter, which includes the topics that people are currently talking about the most. We propose the use of Trending concept in recommender systems as well. In this context, trending items are the current hit interests. These interests can be extracted per category value. For example, if a new hit movie attracted the attention of teens, this would appear as trending interest for them. It should be noted that trending interests are slightly different from the current interests. Trending interests are those that were not interests in the previous period of times, while current interests can persist for a long period of time. These trending interests can be given as a feedback to the recommender systems to improve their accuracy.

Moreover, further studies can be conducted to analyze the relationships among these trending interests and try to find patterns to predict future trending interests. Trending interests may have a specific behavior that comply with, and this would be a an interesting topic to study for certain domains. This would be left for future studies.

To extract trending interests per category value, we adopt the use of df-idf, which is one of the methods that has been used in the literature to extract trending topics. In this method, each term t is given a value of importance, which is the df-idf$_t$ , and it depends on two factors; the frequency of

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the term in the current time slot and the average frequency of the term in the last t time slots.
Its equation is the following.

$$df - idf_t = \frac{df_i + 1}{\log\left(1 + \frac{\sum_{j=i}^{t} df_{i-j}}{t}\right) + 1}, \quad (4)$$

where $df_i$ is the frequency of the term in the current time slot, and $df_{i-j}$ is the average frequency of the term in the last t time slots. Similarly, the trending items per category value can be extracted when the items ordered per that category value are stored based on the time of order. Time slot can be domain dependant. In this work, each month is considered a time slot. Therefore, if there is a significant difference between the current amount of orders for an item and the previous amount of orders, this item is considered a trending item for that category value.

TABLE V.
TRENDING ITEMS PER TEEN EXAMPLE.

| Period Trending Item | Period Label |
|---|---|
| Alice Through the looking Glass | May 2016 |
| Finding Dory | June 2016 |
| The secret life of Pets | July 2016 |
| ... | |

## V. EXPERIMENTS

In order to evaluate the system, we used a synthesized dataset composed of 1000 users and 10000 items. These users are of various age, gender, and marital status values. They rate history contains 5000 rates.

For our experiments, we used an Intel® Xeon® server of 3.16GHz CPU and 2GB RAM, with Microsoft Windows Server 2003 Operating System. Also, we used Microsoft Visual Studio 6.0 to read the dataset and execute the methods. We used the set of categories and values presented in Table VI.

TABLE VI.
USED CATEGORIES AND CATEGORY VALUES.

| Category | Category Values |
|---|---|
| Gender | Male, Female |
| Age | Young, Adult, Senior |
| Marital | Single, Married |

*A) Basic System Evaluation*

The evaluation of the system can be divided into two phases, the online phase and the offline phase.
The online phase is the phase when a user orders or rates an item. The offline phase is the phase that is done periodically to extract statistics of the various categories. This phase is not synchronized with user activities.

*Evaluating Online Phase*

When the user orders or rates an item, the coordinator would receives this transaction. First, it forwards the record according to user type to the concerned agents. As an implementation, each user is represented as a bitmap of all the values in all the categories. Therefore, the forward operation needs O(V), where V represents the number of all values in the system. For each 1 bit in the user bitmap, a record forward operation is done to the corresponding agent value. As the total number of values in the system is commonly not large, this operation is quite fast. In our system, the total number of values is 7 as shown in Table VI.
Upon the receipt of the record by the value agent, a check on the type of the operation is done(order or rate operation), and if it is a rate operation, the rate sign will be added to the record. These two operations are negligible. Either orders or likes matrix is updated as given in equations 1 and 2 respectively. Both operations are prime operations and would be done fast. Next, the term vector representing the item will be retrieved from the *Item Term Matrix* and inserted into the *Transaction Term Matrix*. This operation is a sequential search operation which needs O(I), where I is the number of items.

*Evaluating Offline Phase*

In the offline phase, the items with the maximum values in likes and orders matrix are extracted. These operations need $O(i.\log_i)$, where i is the number of items in the array. Finding the most liked terms from the *Transaction Term Matrix* depends on the number of terms. This operation was studied in our previous work in [30] and proved to be efficient.

*B) Evaluating Unique Item Per Category Value Extractor*

To find items correlated with a specific category value, a *User Item Matrix* needs to be created as previously explained in Table IV. Every time the user orders an item, the matrix needs to be updated. The update process is a prime operation and would be fast.
In order to evaluate the performance of CHI square method, we used subsets of the synthesized dataset of various number of users and number of items. The matrix was 66% sparse. Table7 displays the results. Clearly, the CHI square method is efficient in term of execution time, and even with relatively large number of users and number of items, the execution time was in term of few minutes.

TABLE VII.
THE EXECUTION TIME OF CHI SQUARE METHOD USING VARIOUS NUMBER OF USERS AND NUMBER OF ITEMS.

| Number of Users | Number of Items | CHI Square Execution Time (S.) |
|---|---|---|
| 100 | 1000 | 2.8 |
| 100 | 5000 | 17.5 |
| 500 | 1000 | 4.1 |
| 500 | 5000 | 183 |
| 1000 | 1000 | 8.5 |
| 1000 | 5000 | 337 |

*C)Evaluating Trending Items Per Category Value Extractor*

As explained before, in this method, the ordered items per category value must be stored per time slots. This operation is negligible.

Periodically, to find trending items per category value, equation 4 would be used. Using various subsets of the synthesized dataset with various number of items, we studied the performance of trending item extractor. Table VIII displays the results. Obviously, the time needed to evaluate the items is very fast even with large number of

ISBN: 978-988-14047-1-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

items. Extracting the items with the maximum values would need O(i.log i), where i indicates the number of items.

TABLE VIII.
THE EXECUTION TIME OF THE TRENDING ITEMS EXTRACTOR USING VARIOUS NUMBER OF ITEMS.

| Number of Items | Trending Items Extraction (S.) |
|---|---|
| 500 | 0 |
| 1000 | 0.01 |
| 5000 | 0.04 |
| 10000 | 0.07 |

As for the accuracy, both df-idf and CHI square has been used widely in the literature and proved their accuracy. Please refer to [29] and [30] respectively for details.

## VI. Conclusion

In this work, an agent-based system to extract the interests of user groups per category value is proposed. Mainly, the system is able to extract various types of information such as the most ordered items per category value, the most liked items per category value, the trending items per category value, the uniquely ordered items per category value, and the terms that represent the likes of the groups. Experimental work showed that this system is efficient.

Future work can be done to integrate this part into recommender systems to improve its accuracy. Further studies can be conducted to expand the use of trending items to predict the behavior of the market in the future and to find similarities among markets.

## References

[1] M.J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., vol. 4321, pp. 325-341, Springer-Verlag, 2007.

[2] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 5-53, 2004.

[3] R. Burke, "Hybrid Web Recommender Systems," *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., vol. 4321, ch. 12, pp. 377-408, Springer, 2007.

[4] M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning*, vol. 27, pp. 313-331, 1997.

[5] N. Littlestone and M. Warmuth, "The Weighted Majority Algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212- 261, 1994.

[6] R.J. Mooney, P.N. Bennett, and L. Roy, "Book Recommending Using Text Categorization with Extracted Information," Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998.

[7] S. Robertson and S. Walker, "Threshold Setting in Adaptive Filtering," *J. Documentation*, vol. 56, pp. 312-331, 2000.

[8] E. Rich, "User Modeling via Stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329-354, 1979.

[9] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Comm. ACM*, vol. 35, no. 12, pp. 61-70, 1992.

[10] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. 1994 Computer Supported Cooperative Work Conf.*, 1994.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l WWW Conf.*, 2001.

[12] Y.-H. Chien and E.I. George, "A Bayesian Model for Collaborative Filtering," *Proc. Seventh Int'l Workshop Artificial Intelligence and Statistics*, 1999.

[13] D. Pavlov and D. Pennock, "A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains," *Proc. 16th Ann. Conf. Neural Information Processing Systems* (NIPS '02), 2002.

[14] I. Soboroff and C. Nicholas, "Combining Content and Collaboration in Text Filtering," *Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering*, Aug. 1999.

[15] L.H. Ungar and D.P. Foster, "Clustering Methods for Collaborative Filtering," Proc. Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08 1998.

[16] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners*, L. Rokach, B. Shapira, P. Kantor, and F. Ricci, eds., pp. 217-250, Springer, 2011.

[17] Y.-K. Wang, "Context Awareness and Adaptation in Mobile Learning," *Proc. IEEE Second Int'l Workshop Wireless and Mobile Technologies in Education (WMTE '04)*, 2004, pp. 154-158.

[18] A. Dey, G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *HumAN-Computer Interaction*, vol. 16, pp. 97-166, Dec. 2001.

[19] L. Nguyen and P. Do, "Learner Model in Adaptive Learning," *World Academy of Science Eng. and Technology*, vol. 45, pp. 395-400, 2008

[20] P. Vashisth and P. Bedi, "Interest-Based Personalized Recommender System," *World Congress on Information and Communication Technologies*, 2011.

[21] Y. Z. Wei, L. Moreau, and N. R. Jennings, "Learning Users' Interests by Quality Classification in Market-Based Recommender Systems," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1678-1688, Dec. 2005.

[22] G. Aghili, M.. Shajari, S. Khadivi, and M. A. Morid, "Using Genre Interest of Users to Detect Profile Injection Attacks in Movie Recommender Systems," *Proc. 10th Ann. Conf. Machine Learning and Applications*, 2011.

[23] S. Phuvipadawat and T.Murata, "Breaking news detection and tracking in Twitter," *Proc.Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM Int. Conf.*, 2010, vol. 3, pp. 120–123.

[24] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory search and topic summarization for Twitter," in *ICWSM*, W. W. Cohen, S. Gosling, W. W. Cohen, and S. Gosling, Eds. Palo Alto, CA, USA: AAAI Press, 2010.

[25] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on Twitter," *Proc. ICWSM: 5th Int AAAI Conf. Weblogs and Social Media*, 2011.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[27] D.M. Blei and J. D. Lafferty, "Dynamic topic models," *Proc. ICML: 23rd Int. Conf. Machine Learning*, New York, NY, USA, 2006, pp. 113–120, ACM.

[28] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Peaks and persistence: Modeling the shape of microblog conversations," *Proc. CSCW: ACM Conf. Computer Supported Cooperative Work*, New York, NY, USA, 2011, pp. 355–358.

[29] J. Weng and B.-S. Lee, "Event detection in Twitter," *Proc. 5th Int.Conf. Weblogs and Social Media*. Palo Alto, CA, USA: AAAI Press, 2011.

[30] B. Hawashin, A. Abusukhon, A. Mansour, "An Efficient User Interest Extractor for Recommender Systems," *Proc. of the World Congress on Engineering and Computer Science*, 2015.