Combining social and semantic information for recommandation : comparative study

Dalia Sulieman[1], **Maria Malek**[2], **Hubert Kadima**[3], **Dominique Laurent**[4]

1. ETIS/ENSEA Université de Cergy-Pontoise CNRS France

2. EISTI Cergy-Pontoise France

ABSTRACT. *In this paper we present algorithms for recommender systems. Our algorithms rely on a semantic relevance measure and social network centrality measures to partially explore the network using depth-first search and breath-first search strategies. We implement and compare several social network centrality measures. We apply our algorithms on real dataset: the MovieLens one. Our results show that having algorithms combining degree and betweenness give high precision and recall values. Moreover, the importance of our algorithms rely on the fact that these algorithms explore a small part of the graph instead of exploring all the graph as the classical searching methods do.*

RÉSUMÉ. *Dans cet article nous présentons des algorithmes pour la recommandation dans les réseaux sociaux. Ces algorithmes combinent, les mesures centralités dans les réseaux sociaux et les profils sémantiques des utilisateurs dans le processus de l'élaboration de la recommandation. Nous intégrons des heuristiques dans l'exploration de graphe (parcours en profondeur DFS et parcours on largeur BFS). Nous avons appliqué ces algorithmes sur un ensemble de données réelles extraits des données de MovieLens. Nos résultats montrent des valeurs de précision, de rappel et de F-measure satisfaisantes.*

KEYWORDS: *recommender systems, social network analysis, graph algorithms, semantic taxonomy.*

MOTS-CLÉS : *systèmes de recommandation, analyse de réseaux sociaux, algorithmes de graphe, taxonomie sémantique.*

DOI:10.3166/HSP.4.1-11

1. Introduction

It is well known that the information available on the Web increases rapidly in time. As a result, humans are not capable of understanding, exploiting or even handling such a huge amount of information. *Recommender systems* are thus widely used to overcome this information overload, by filtering information in order to help users in making choices according to their interests. Moreover, it is now common practice that users be connected through a *social network*, in which vertices and edges represent respectively people and their social interactions (such as friendship and co authorship) (Newman, 2010).

In this paper we introduce a *semantic social recommender system*, in which a set of users and a set of items such that users are connected through a social network, and users and items are described via a taxonomy. In this setting, given an item, we use a depth first search and breadth first search algorithms to search the social network in order to compute a relevant set of users to whom the item can be recommended, while scanning the network as few as possible. Thus our main contribution is to provide algorithms that combine all available information (namely, the taxonomy relating items, users preferences seen a part of that taxonomy and the social network connecting users) in order to efficiently compute the set of users to whom a given item should be recommended. Moreover, we test and compare the effects of several types of centrality measures, mainly Degree Closeness and Betweenness, on the recommender system. Furthermore, we suggest to combine Degree and Betweenness centrality in one algorithm. The experiments reported in the paper show that our algorithms have good accuracy and high performance.

2. Related Work

In this section we briefly review related work on collaborative filtering (social) recommender systems, and semantic-social recommender systems. Mainly, collaborative filtering recommender systems are user-based (Resnick *et al.*, 1994), item-based, or even graph-based using graph searching algorithms, such as: shortest path (Mirza *et al.*, 2003), random walk (Jamali, Ester, 2009) and breadth first search (Aggarwal *et al.*, 1999). Moreover, semantic-social recommender systems combine collaborative filtering with semantics about users, these semantics could be: user tags (Julia Stoyanovich, Sihem Amer Yahia, Cameron Marlow and Cong Yu, 2008), clusters of user tags (Shepitsen *et al.*, 2008), keyword user profiles (Malek, Sulieman, 2010) or taxonomy user profiles (Ziegler *et al.*, 2004).

Proposed in 1994 by Grouplens (Resnick *et al.*, 1994), Newsnet is one of the earliest collaborative filtering recommender systems. Newsnet is a user-based collaborative filtering, based on the Pearson r correlation coefficient. Ringo (Shardanand, Maes, 1995) is another collaborative filtering recommender system, which uses personalization to recommend music and artists. In 1999, IRA (Intelligent Recommendation Algorithm, (Aggarwal *et al.*, 1999)) was proposed as a graph-based collaborative filtering recommender system, in which a breadth-first search algorithm is used to compute the shortest paths between graph vertices (users). Moreover, user-item bipartite graph and one-mode projection are used in a movie recommender system proposed in (Mirza *et al.*, 2003). In this system a recommendation graph is defined as the sum of the bipartite graph and the one-mode projection graph, then a shortest path algorithm is applied on the recommender graph in order to compute the recommendation. In (Jamali, Ester, 2009), a random walk algorithm is proposed to recommend items in a trust network. This algorithm recommends items based on ratings expressed by trusted friends, using random walk and probabilistic item selection.

Other recommender systems include semantic aspects, in addition to collaborative filtering aspects. In (Julia Stoyanovich, Sihem Amer Yahia, Cameron Marlow and Cong Yu, 2008) a recommendation algorithm is introduced for collaborative URL tagging. In this system, user interests are modeled according to their social ties and the vocabularies they use to tag URLs.

In (Shepitsen *et al.*, 2008) similar tags are grouped in clusters, these tag clusters are used as intermediate sets between users and items, in fact the recommendation is based on two closeness values: closeness between users and tag clusters, and closeness between item and the tag clusters.

In (Malek, Sulieman, 2010) an algorithm is proposed to give recommendations in a professional social network. The goal of this algorithm is to recommend a group of ranked authors similar to a given criterion, in a social network of connected authors. In this system, recommendations are based on the social network analysis and the user profiles, these profiles are represented by vectors of keywords. In (Ziegler *et al.*, 2004) the authors proposed to represent the users by a vector of interest scores assigned to topics taken from domain taxonomy; this taxonomy represents item categories.

3. Recommender System For Social Networks

3.1. Taxonomy User Profile

User profile contains all the possible information about any given user, such as activities and interests. User profile has several types of representation as, taxonomy representation (Gauch *et al.*, 2007). Moreover, taxonomy is defined as a collection of entities which are organized into a hierarchal structure, named "is-a" hierarchy, in order to describe certain objects in a certain domain. However, several recommender systems use taxonomy because taxonomy representation of information is a very helpful tool to estimate users preferences especially in the case of lack information about users. In our algorithms, we use taxonomy to represent both user and item preferences. See Figure 1.

3.1.1. Semantic Similarity Measure

Semantic similarity measures are used to compute the closeness between any pairs of taxonomy concepts (Resnik, 1995). In our model we use semantic similarity measure to find out the relevancy between the input item and user profile. This measure is described in (Sulieman *et al.*, 2013) and it is given by Equation 1

$$\sigma(U, I) = \frac{1}{\mu} \left(\sum_{(c,l) \in U \cap I} l \right) \tag{1}$$

Where U is the user preferences and I is the item preferences. (c, l): c is one concept of the preferences and l is the level of this concept. And $\mu = \max \left(\sum_{(c,l)\in U} l, \sum_{(c,l)\in I} l \right)$.

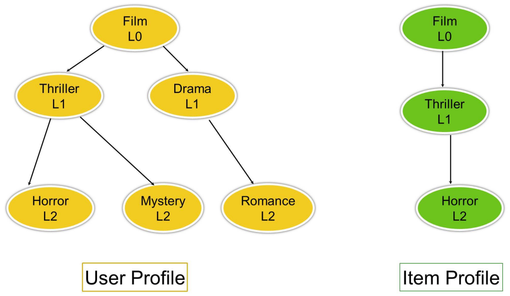

Figure 1. Taxonomy profiles

3.2. Social Network Analysis Measures: Centrality Measures

Freeman has classified centrality measures, according to two aspects: vertex location in the network and network structure, into three main categories: *Degree, Closeness* and *Betweenness* (Freeman, 1979). However, these measures are become the basics of social network analysis, where almost all the other centrality measures are derived from these three ones.
Lately, Borgatti has characterized Degree, Closeness and Betweenness according to three aspects: The vertex involvement in the network structure (S. Borgatti, Everett, 2006), The topology of the network flows and the methods of information spread in the network (S. P. Borgatti, 2005).

In this subsection we present the definition of Degree, Closeness and Betweenness centralities in order to clarify how we employ them in our proposed recommendation algorithms.

3.2.1. Degree Centrality

Degree is the simplest and the most intuitive graph centrality measure (Newman, 2010). It is calculated by counting the number of direct ties of any given vertex in the graph (Freeman, 1979). Degree centrality of any given vertex v_k in the graph $G(V, E)$ is given by Equation 2:

$$C_D(v_k) = \sum_{i=1}^{n} e(v_i, v_k) \tag{2}$$

$e(v_i, v_k) = 1$ if and only if v_i and v_k are adjacent, and 0 otherwise.
In any network, vertex importance depends on the number of its direct connections.

3.2.2. Closeness Centrality

Closeness centrality is remarkably used to calculate the distances (number of edges) connecting any pair of vertices in the graph. Obviously, Closeness centrality plays an important role in terms of information diffusion and transmission costs.
Closeness centrality of a given vertex v_k in $G(V, E)$ is defined as the inverse value of the sum of the length of the shortest paths linking this vertex with all the other graph's vertices (Freeman, 1979; S. P. Borgatti, 2005; Newman, 2010). And it is given by Equation 3:

$$C_C(v_k) = \frac{1}{\sum_{j=1}^{n} d(k, j)} \tag{3}$$

$d(i, j)$ is the number of edges in the shortest path connecting v_i and v_j.

3.2.3. Betweenness Centrality

Betweenness centrality is an important measure which treats the vertices as mediators, and determines their impact in the communications between other vertices. It calculates the number (frequency) of the shortest paths passing through a given vertex v_k in the graph $G(V, E)$ (Freeman, 1979). Betweenness is given by Equation 4:

$$C_B(v_k) = \sum_{i<}^{n} \sum_{j}^{n} b_{ij}(v_k) \tag{4}$$

$i \neq j \neq k$, n: vertices number, $b_{ij}(v_k)$: the probability of v_k falls between v_i and v_j.
Both *Degree* and *Closeness* centralities, are used to measure the vertex ability to control information spread and message transmission. While *Betweenness* centrality is used to measure the vertex ability to facilitate the information flow in the graph passing through vertices. Therefore, Betweenness is not a measure of how a vertex is well connected, but it is a measure of how much a vertex falls in the shortest paths passing between vertices. As consequence, a vertex could have low degree and closeness values, but still have high betweenness.

3.3. Graph Searching Algorithms

In this section we briefly describe two graph searching algorithms "Depth First Search DFS" and ”Breadth First Search”. These two algorithms are fundamental and the majority of important graph searching algorithms are deviated from them such as: Dijsktra and A* algorithms.

3.3.1. Depth First Search DFS algorithm

For a long time DFS, has been considered as one of the fundamental graph searching algorithms (Tarjan, 1972). DFS searches deeper in the graph whenever it is possible. It explores the graph edge out of the most recently discovered vertex v that still has unexplored edges leaving it. Once all of the v edges have been explored, the search backtracks to explore edges leaving the vertex from which v was discovered. This process continues until discovering all the vertices that are reachable from the original source vertex. If any undiscovered vertices remain, then DFS selects one of them as a new resource, and it repeats the search from that source. The algorithm repeats this entire process until discovering all the graph vertices (Cormen *et al.*, 2009).

3.3.2. Breadth First Search BFS Algorithm

It is one of the simplest and basic graph searching algorithms. For a given graph $G(V, E)$ and a distinguished source vertex v_k, BFS explores the vertices that are reachable for v_k, and computes the distance from v_k to these vertices. BFS explores all the vertices at distance d from the source v_k, then it explores the vertices at distance $d+1$. The algorithm repeats these steps until visiting all the graph vertices (Cormen *et al.*, 2009).

4. Recommendation Algorithms

Our proposed algorithms are based on three main concepts: *Graph Searching algorithms, Social Network Analysis Measures and Semantic Relevancy* between the input item and the user.

DEFINITION 1 (The Bipartite Graph G=(V,E)). — *Let I be a set of items, and let U be a set of users. The bipartite graph is described as the graph $G = (V, E)$. $V(G)$ is a set of $G(V, E)$ vertices, where $V = I \bigcup U$ and $I \bigcap U = \emptyset$, and $E(G)$ is a set of $G(V, E)$ edges where $\forall e(v_i, v_j) \in E(G)$ if $v_i \in I$ then $v_j \in U$ and if $v_j \in U$ then $v_i \in I$.*

DEFINITION 2 (Users Collaboration Graph $G(U, E)$). — *Users collaboration graph is a users one-mode projection weighted graph $G(U, E)$. $\forall u, u' \in U(G)$, an edge $e(u, u')$ is created if and only if there is at least one item $I \in V(G)$, from the bipartite graph $G(V, E)$, in common between u and u'. Moreover, edge are weighted by the number of common items between users.*

DEFINITION 3 (Recommendation Query). — *Let $I(x)$ be an item, described by taxonomy. And let $G = (u, E)$ the users one-mode projection of the MovieLens bipartite graph $G(V, E)$. The recommendation query asks the system to search the collaboration network in order to find the users that could like the item.*

4.1. Algorithm Details

In our approach, we search any collaboration social network in which vertices are users and edges are weighted by the number of common items between these users. We apply heuristic depending on two mesures: the semantic similarity (between the users and the input item) and the vertex (user) centrality mainly (Degree, betweenness and closeness). This heuristic gives the possibility to avoid exploring all the

vertices and all the edges in the social network. So, the algorithms start the search from the vertices having high centrality (Degree, Closeness, Betweenness, or Degree and Betweenness) values and explore the graph either in depth-first search manner or in breadth-first search manner. Moreover, the algorithms continue processing the starting vertex successors if and only if they satisfy the semantic similarity and the centrality requirements.
The *Algorithm Input:* is an item I to be recommended to users $u_i \in U(G)$, the item I is attached to a taxonomy profile.
Algorithm Output: A recommendation list contains all the users who could be interested by the input item $I(x)$.
The algorithms are described by the following steps:

1. Initialization: we order all the vertices according to their centrality values, then we stock them into a centrality vector (we do not add all the vertices, we only add the ones with the highest centrality values).

2. Source Vertex: As centrality vector contains the most important graph vertices, DFS starts the search from these vertices (vertex by vertex).

3. Depth First Search DFS: we apply DFS algorithm with heuristic. For this heuristic we propose to combine semantic with social characteristics. Semantic characteristics depend on the semantic relevancy between users and input item, as described in 3.1.1. While, social characteristics depend on the centrality of these users, as discussed in 3.2. It is remarkably noted that according to the heuristic: DFS passes only through vertices that satisfy semantic and centrality criteria.

4. Stopping the Search: The algorithm stops graph searching in only two cases: if the centrality vector is empty or if DFS has visited all the vertices.

We repeat the same steps with another algorithm "Breadth First Search Algorithm". In order to implement and test the performance of the two algorithms DFS with semantic social heuristic and BFS with semantic social heuristic.

Furthermore, we employ Degree, Closeness and Betweenness centralities in these two algorithms, by experimenting DFS and BFS four times: firstly using Degree centrality, secondly using Closeness centrality, thirdly using Betweenness centrality and lastly by combing Degree and Betweenness centrality in one algorithm.

5. Experiments and Results

5.1. Data Set: MovieLenes

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota [1]. This data set consists of: $100,000$ratings $(1-5)$ from 943 users on 1682 movies. Each user has rated at least 20 movies. Simple demographic info about the users (age, gender, occupation, zip) are given. Movies are described via

1. http://www.grouplens.org

their geners (Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western). We build MovieLens collaboration social network as described in definition 2. As a result, we obtain a social network with 999 vertices and $136,365$ edges.

5.2. Evaluation: Metrics

In our experiments, we use the *precision*, *recall* and *F-measure* to evaluate the accuracy of the recommendation algorithms (Herlocker *et al.*, 2004). Algorithms are also assessed according to their performance.

5.2.1. Accuracy Metrics

Precision is defined as the probability that a selected user is relevant, and recall is defined as the probability that a relevant item is selected. We recall below the definitions of these two measures where TP (true positive) is the number of relevant users who have been recommended, TN (true negative) is the number of relevant users who have not been recommend, and FP (false positive) is the number of irrelevant users, who have been recommend. Also we use F-measure which equals to two times the multiplication of precision and recall divided by their sum.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + TN}$$

5.2.2. Performance

Algorithms performance is computed by finding the percentage of graph vertices (users) who have been visited by the recommendation algorithms, in order to find the recommended users.

5.3. Algorithms Comparision and Discussion

Figures 2 and 3, show that DFS algorithm has better precision and recall values than BFS. But, it is remarkably that these differences in accuracy measures are not very big. Moreover, according to Figure 4 DFS shows better performance than BFS. Because, BFS visits almost all the graph vertices while DFS visits less than 25% of the graph vertices.

Degree centrality gives good accuracy values, because the higher degree centrality means the more influential role of the vertex in the network. Moreover, the differences between degree and closeness centrality accuracy values are not very big because, closeness and degree has the same nature and the higher closeness means the higher possibility the vertex be reachable from other vertices. So, in the case of recommender system it is easy to find relevant users if we start the search from vertices with high degree or closeness. Moreover, the accuracy has no good values in the case of using betweenness centrality because betweenness is used to control information flow through the network and to measure the vertex ability to control information spread and message transmission. So, betweenness centrality in such recommender system is not useful, but as we combine both Degree and betweenness centralities we can improve algorithms accuracy by benefiting from the characteristics of Degree and Betweenness centralities.

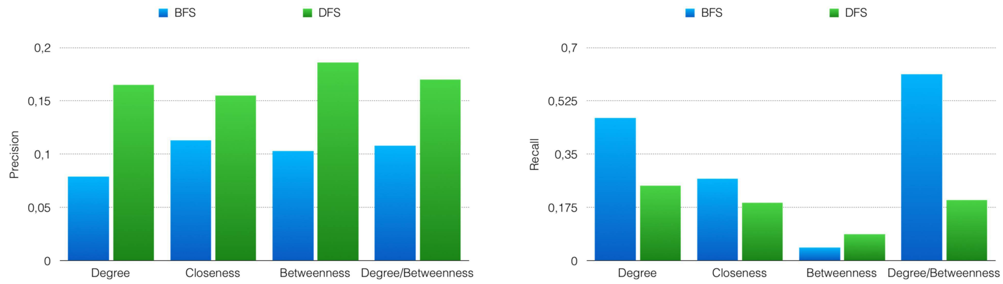

Figure 2. Precision and Recall values

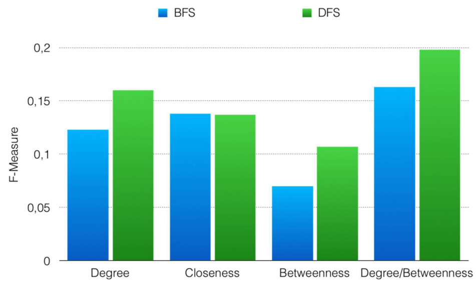

Figure 3. F-measure values

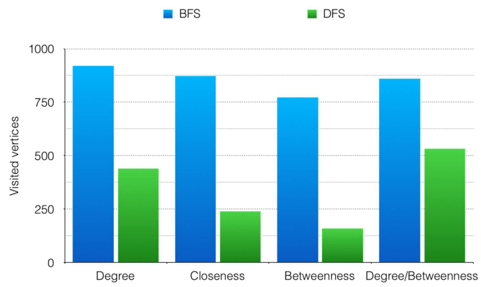

Figure 4. The maximum number of explored vertices

6. Conclusion

Nowadays, social networks become one of the most important resource of information, and sometimes it is very difficult to search these networks. However, searching social networks could not be only affected by the social network characteristics, such vertices centralities, but also information about its users, such as interests and relations.

In this paper we introduced two semantic social recommendation algorithms based on DFS and BFS strategies, these algorithms recommend an input item to a group of users. In fact, we assume that, users are linked via collaboration social network, and users and items are described via semantic taxonomy. Our proposed algorithms are mainly based on Depth First Search and breadth first search algorithms with some modifications, which are related to firstly, the semantic relevancy between users and the input item ; and secondly, the social network analysis measures mainly Degree, Closeness and Betweenness. We applied these algorithms on a real dataset (MovieLens Dataset). Our results showed that there is no big differences between DFS and BFS in the terms of accuracy, while the performance of DFS is better than BFS because DFS sometimes searches less than 25% of the dataset. In fact, searching a small amount of the dataset and having good accuracy values is not found in the other classical recommendation methods. Moreover, we compared the utilization of Degree, Betweenness and Closeness centralities and we found that: combining Degree and Betweenness centralities in one algorithm could improve the algorithm accuracy.

References

Aggarwal C. C., Wolf J. L., Wu K., Yu P. S. (1999). Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In *Kdd '99: Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining*, pp. 201–212. San Diego, California, United States, ACM.

Borgatti S., Everett M. (2006, October). A Graph-theoretic perspective on centrality. *Social Networks*, Vol. 28, No. 4, pp. 466–484.

Borgatti S. P. (2005, January). Centrality and network flow. *Social Networks*, Vol. 27, No. 1, pp. 55–71.

Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. (2009). *Introduction to Algorithms* (third edition ed.). The MIT Press. Hardcover.

Freeman L. (1979). Centrality in social networks conceptual clarification. *Social Networks*, Vol. 1, No. 3, pp. 215–239.

Gauch S., Speretta M., Chandramouli A., Micarelli A. (2007). User Profiles for Personalized Information Access The Adaptive Web. In P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *The adaptive web*, Vol. 4321, pp. 54–89. Berlin, Heidelberg, Springer Berlin / Heidelberg.

Herlocker J. L., Konstan J. A., Terveen L. G., Riedl J. T. (2004, January). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, Vol. 22, No. 1, pp. 5–53.

Jamali M., Ester M. (2009). Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining*, pp. 397–406. NY, USA, ACM.

Julia Stoyanovich, Sihem Amer Yahia, Cameron Marlow and Cong Yu. (2008). A study of the benefit of leveraging tagging behavior to model usersínterests in del.icio.us. In *In aaai spring symposium on social information processing.*

Malek M., Sulieman D. (2010, September). Exhaustive and guided algorithm for recommendation in a professional social network. In *7th conference on application of social network analysis*, pp. 15–17. Zürich.

Mirza B. J., Keller B. J., Ramakrishnan N. (2003, March 1). Studying Recommendation Algorithms by Graph Analysis. *Journal of Intelligent Information Systems*, Vol. 20, No. 2, pp. 131–160.

Newman M. (2010). *Networks an introduction*. Oxford University Press.

Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Cscw '94: Proceedings of the 1994 acm conference on computer supported cooperative work*, pp. 175–186. New York, NY, USA, ACM.

Resnik P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Ijcai'95: Proceedings of the 14th international joint conference on artificial intelligence*, pp. 448–453. San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.

Shardanand U., Maes P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of acm CHI'95 conference on human factors in computing systems*, Vol. 1, pp. 210–217.

Shepitsen A., Gemmell J., Mobasher B., Burke R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 acm conference on recommender systems*, pp. 259–266. New York, NY, USA, ACM.

Sulieman D., Malek M., Kadima H., Laurent D. (2013). Exploiting semantic and social information in recommendation algorithms. In *Workshop on information search, integration and personalization (isip)*, p. 10. Sapporo, Japan, Springer Verlag.

Tarjan R. (1972). Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, Vol. 1, No. 2, pp. 146–160.

Ziegler C. N., Lausen G., Lars S. T. (2004). Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth acm international conference on information and knowledge management*, pp. 406–415. New York, NY, USA, ACM.