



# A fine-grained social network recommender system

Markos Aivazoglou<sup>1</sup> · Antonios O. Roussos<sup>1</sup> · Dionisis Margaris<sup>2</sup> · Costas Vassilakis<sup>3</sup> · Sotiris Ioannidis<sup>1</sup> · Jason Polakis<sup>4</sup> · Dimitris Spiliotopoulos<sup>3</sup> 

Received: 18 April 2019 / Revised: 9 December 2019 / Accepted: 16 December 2019 / Published online: 19 December 2019  
© Springer-Verlag GmbH Austria, part of Springer Nature 2019

## Abstract

Recommender systems have greatly evolved in recent years and have become an integral part of the Web. From e-commerce sites to mobile apps, our daily routine revolves around a series of “small” decisions that are influenced by such recommendations. In a similar manner, online social networks recommend only a subset of the massive amount of content published by a user’s friends. However, the prevalent approach for the content selection process in such systems is driven by the amount of interaction between the user and the friend who published the content. As a result, content of interest is often lost due to weak social ties. In this paper, we present a fine-grained recommender system for social ecosystems, designed to recommend media content (e.g., music videos, online clips) published by the user’s friends. The system design was driven by the findings of our qualitative user study that explored the value and requirements of a recommendation component within a social network. The core idea behind the proposed approach was to leverage the abundance of preexisting information in each user’s account for creating interest profiles, to calculate similarity scores at a fine-grained level for each friend. The intuition behind the proposed method was to find consistent ways to obtain information representations that can identify overlapping interests in very specific sub-categories (e.g., two users’ music preferences may only coincide on hard rock). While the system is intended as a component of the social networking service, we developed a proof-of-concept implementation for Facebook and explored the effectiveness of our underlying mechanisms for content analysis. Our experimental evaluation demonstrates the effectiveness of our approach, as the recommended content of interest was both overlooked by the existing Facebook engine and not contained in the users’ Facebook News Feed. We also conducted a user study for exploring the usability aspects of the prototype and found that it offers functionality that could significantly improve user experience in popular services.

**Keywords** Social networks · Recommender systems · Content analysis · Interest profiles · Usability

## 1 Introduction

The enormous growth of available content in current applications, and web applications in particular, has created information overload for the application users. This necessitated

the use of personalization techniques to alleviate this problem (Berkovsky and Freyne 2015), by prioritizing or limiting information presented to the users according to its perceived value for them. With the advent of social networks, such as Facebook (Facebook 2019) and Twitter (Twitter 2019), used by millions of people every day, large volumes of data generated by these networks are widely available. Researchers seek methods to exploit these data for personalization purposes (Bakshy et al. 2012; Cai et al. 2010; He and Chu 2010). Such data are deemed of high value in the context of personalization, because of the importance and the intrinsic relationship with people’s everyday lives (Margaris et al. 2018). One of the effects of the overwhelming amount of the generated information is that users spend longer times to sort through their news feeds to track the interesting content. Friend-generated content is an important part of the

---

✉ Dimitris Spiliotopoulos  
dspiliot@uop.gr

<sup>1</sup> Institute of Computer Science, Foundation for Research and Technology, Iraklio, Greece

<sup>2</sup> Department of Informatics and Telecommunications, University of Athens, Athens, Greece

<sup>3</sup> Department of Informatics and Telecommunications, University of the Peloponnese, Tripoli, Greece

<sup>4</sup> Computer Science Department, University of Illinois at Chicago, Chicago, USA

social interaction between users on social networks. However, the massive amount of content published by friends in social networks (Facebook news feed 2016) can result in users overlooking content that might be of interest, due to the overall noise. Furthermore, the content presented in the Facebook News Feed is heavily influenced by the user's relationship to the friend that published it (Facebook 2019). As a result, content that might be of great interest to the user may be omitted due to a "weak" social tie to the content publisher. Two main challenges arise from the above:

1. How do users of social networks receive the importance of social ties on recommended content?
2. How much of such information remains invisible, how can it be revealed, and what impact could it have to the quality of the user experience?

This work addresses the aforementioned challenges by proposing a recommender system that can exist within the social network, categorize content and offer suggestions based on the interest similarities between the user and the friend that published the content.

The main contributions of this work are the following:

1. We explore and report on the findings on the perception of Facebook users on the value of social recommendations within their friend network and their expectations regarding their social circle's ability to provide media content recommendations, by conducting a qualitative user study.
2. Based on the findings of the user study, and using them to derive the requirements, we design, develop and evaluate a method to recommend the most relevant and interesting media published by a user's social graph.
3. We developed a prototype implementation that leverages sentiment analysis and named entity recognition for identifying and labeling content "approved" by the user's friends; content is ranked based on the similarity evaluation of fine-grained interest profiles automatically derived from each user's social data.

Our study affirms our driving hypothesis that users are aware of their overlapping interests with a subset of their social circle and have high expectations regarding the relevance and interest of posted content. Our study also identifies specific requirements for recommender system design, aimed to meet user expectations and identify the parameters that lead to successful social recommendation for media content within a network of friends. We conducted a pilot study with real Facebook users in order to assess the value of the work proposition and explore the effectiveness of unobtrusive social recommendations in a practical setting. Our method was evaluated over a period of 2 weeks using

real-time user data, with user feedback validating both the effectiveness of our content selection approach and the overall usability of the prototype system.

This paper extends the work presented in Aivazoglou et al. (2017) on the following key aspects:

1. On the system level, the full system architecture that was implemented and the component analysis (Sects. 4.1–4.3).
2. On the application level, the Facebook application developed for both a preliminary exploratory study for the collection of requirements and validation of the key motivation factors and the evaluation of the proposed approach (Sect. 4.5).
3. On the methodology and work, the results of an extensive usability evaluation of the proposed approach using real users to measure effectiveness of the proof-of-concept implementation heuristics and algorithms (Sect. 5).

The rest of the paper is structured as follows: Sect. 2 overviews the related work, Sect. 3 defines the problem to be addressed and presents the user study and motivation, while Sect. 4 presents the methodology and architecture of the proposed system. Section 5 presents the evaluation of the proposed approach. Section 6 presents the conclusion and the future work.

## 2 Related work

With the advent of social networks, social network recommender systems have received considerable research attention. Chamoso et al. (2018) propose a relationship recommender system for a business and employment-oriented social network. The presented system functions by extracting relevant information from the social network which it then uses to adequately recommend new contacts and job offers to users. The recommender system uses information gathered from job offer descriptions, user profiles and users' actions. Then, different metrics are applied in order to discover new ties that are likely to convert into relationships. Margaritis and Vassilakis (2018) propose a social network recommender system that exploits temporal information from the user rating database to identify periods that social network users have not submitted new ratings, which are termed as rating abstention intervals. The presence of rating abstention intervals is shown to be positively associated with a shift of interest and, therefore, can provide the basis for amplifying or attenuating the weight assigned to social network user ratings in the recommendation generation process. Ma et al. (2018) propose a novel decentralized framework, namely ARMOR, which utilizes online social network users' social attributes and trust relationships to achieve the friend recommendation in a privacy-preserving manner.

In numerous previous studies, social network information is used in order to alleviate problems that affect recommender systems, such as cold-start and item sparsity (Camacho and Alves-Souza 2018). Contrates et al. (2018) propose a recommendation process that includes sentiment analysis to textual data extracted from Facebook and Twitter and present results of an experiment in which this algorithm is used to reduce the cold-start issue. Jones et al. (2017) suggest an ontological sub-matrix factorization-based approach for recommending items to a new social network user, however without the extraction of user personal information, thereby respecting his privacy.

Mohammadi and Andalib (2017) propose the usage of a new measurement, namely opinion leaders, in order to alleviate the cold-start problem. Opinion leader is a person that his opinion has an impact on the target user. As a result, in the case of a new user logging in and the user-item's matrix sparseness, the opinion of opinion leaders can be used to offer the appropriate recommendation for new users and thereby increase the accuracy of the recommender system. Reshma et al. (2016) propose a new approach which predicts the ratings of items by considering directed and transitive trust with timestamps and profile similarity from the social network along with the user-rated information.

Another issue explored extensively is how a social network recommender system can determine trust and influence between social network users and exploit this information in order to improve recommendations quality to its users. In this direction, Zhang et al. (2018) simulate the social influence diffusion in the social network in order to find the global and local influence nodes and then embed this dual influence data into a traditional recommendation model to improve accuracy. Mathematically, they formulate the global and local influence data as new dual social influence regularization terms and embed them into a matrix factorization-based recommendation model. Margaritis et al. (2016) enhance recommendation algorithms used in social networks by taking into account qualitative aspects of the recommended items, such as price and reliability, the influencing factors between social network users, the social network user behavior regarding their purchases in different item categories and the semantic categorization of the products to be recommended. Wang et al. (2018) model the influence spread process as the fluid update process in three dimensions: the fluid height difference, the fluid temperature and the temperature difference. Moreover, they formulate the Maximizing Positive Influenced Users (MPIU) problem and design a greedy algorithm, namely Fluidspread, in order to solve it. Kalai et al. (2018) present a Web service decentralized discovery approach which is based on two complementary mechanisms. The trust detection is the first mechanism, used to detect the social trust level among users, while the service recommendation is the second mechanism which

combines the social and collaborative approaches to recommend to the active user the appropriate services according to the expertise level of his most trustworthy friends. De Meo et al. (2018) propose an approach to extend global reputation models with a local reputation, computed on the ego network of the user, by means of an unsupervised approach. Their model considers the different relevance given to local and global reputation, the threshold that is used to consider a user unreliable and the dimension of the user's ego network. Margaritis et al. (2019) propose a simple, yet effective algorithm that combines limited collaborative filtering information of user ratings and user social relations (friendship, trust, etc.), aiming to consistently improve both the rating prediction quality and prediction coverage (i.e., the percentage of the cases for which a personalized prediction can be computed).

The supplemental use of text reviews is also proposed, in order to tell whether a user is expressing likeness or disaffection on a topic. Integrating opinion mining in recommender systems could provide more accurate information about users' moods, which could be a significant improvement to topic rating. Pasricha and Solanki (2019) propose a book recommendation approach based on identification of opinion leaders in detected communities of social networks by using information related to their interests, preferences, age and online available attributes on social networks. Tewari et al. (2019) propose an approach which uses opinion mining that analyzes reviews and extracts different products features. Furthermore, the proposed approach finds users' inclination toward different features of products and based on that analysis it recommends products to users. Da'u et al. (2020) propose a weighted Aspect-based Opinion mining using Deep learning method for Recommender system (AODR) that can extract product's aspects and the underlying weighted user opinions from the review text using a deep learning method and then fuse them into extended collaborative filtering technique for improving a recommender system. Zheng et al. (2018) propose a tourism destination recommender system that employs opinion mining technology to refine user sentiment and make use of temporal dynamics to represent user preference and destination popularity drifting over time. These elements are then fused with the SVD++ method by combining user sentiment and temporal influence. Li and Yang (2018) adopt one lexicon-based opinion mining method to extract opinions hidden in reviews and also combine opinions with actual ratings. Additionally, they embed deep neural networks model which breaks through the limitation of traditional collaborative filtering. Shen et al. (2019) propose the sentiment-based matrix factorization with reliability (SBMF + R) algorithm to leverage reviews for prediction. Firstly, they develop a sentiment analysis approach using a new star-based dictionary construction technique to obtain the sentiment score. Secondly, they design a user

reliability measure that combines user consistency and the feedback on reviews. As a final step, they incorporate the ratings, reviews and feedback into a probabilistic matrix factorization framework for prediction. Sangeetha and Prakash (2019) propose the novel product recommendation framework (NPRF) for the prediction of overall opinion and estimate the rating of the product based on the user reviews. Initially, they preprocess the set of large size customer reviews in order to extract the relevant keywords with the help of stop word removal, PoS tagger, slicing and the normalization processes. SentiWordNet library database is then applied to categorize the keywords which are in the form of positive- and negative-based polarity. After extracting the related keywords, the Inclusive Similarity-based Clustering (ISC) method is performed to cluster the user reviews based on the positive and negative polarity.

None of the above-mentioned works considers the issue of fine-grained social recommendations that take into account both user interests and user feedback as recommendation weights and validation factors, respectively.

### 3 Problem definition, user study and motivation

Apart from human interaction in social networks being naturally limited by time constraints, Wilson et al. (2012) found that for the vast majority of users, 70% of their interactions is directed to only 20% of their friends. Viswanath et al. (2009) found that only 30% of Facebook user pairs interact consistently at least once every month. These findings support the motivation for this work, as users may frequently miss posts of interest due to the current content selection approach which relies on social ties, as well as the high speed and volume of posts that hinder users from manually sifting through every friend's profile. Furthermore, De Pessemier et al. (2011) found that there is a positive correlation between the popularity of user-uploaded videos and the size of the user's social circle, indicating that users with many friends can further skew the content recommended in Facebook, as content selection is biased toward popular posts. To obtain further insight regarding the desired characteristics of social recommendations, we conducted a user study with  $N=38$  participants (age = 22–34, 79% male) that provided feedback regarding their interaction, expectations and needs from Facebook, focusing on recommendations from their friends. The goal of this exploratory study was to obtain qualitative information on how Facebook users perceive their time spent on Facebook, across multiple dimensions that describe the use of the social network as a recommendation means for entertainment value (specifically for music and movies). Additionally, specific user relations were explored in order to gain insight into the potential benefit of using social data

in the context of movie and music recommendation. For the latter, our study was designed to collect information-specific requirements for our proposed approach.

#### 3.1 Procedure

The participants were unknown Facebook users that were recruited through the institution mailing lists. They were requested to browse their Facebook News Feed and respond to standard category-establishing questions, such as "Do you use Facebook to find posts about movies/music of interest?". Baseline user profiling was also established using standard methods (interest in entertainment recommendations, time spent on social media, number of friends, etc.) The data were collected via online forms and a follow-up remote debriefing using validation questions to address possible false positives and solidify true positives.

#### 3.2 User profiling and general information

In order to better understand the user interaction and expectations for the proposed solution, the participants were requested to provide feedback on the way that they perceive their social media community on several levels. The initial finding was that the number of friends and time spent on the News Feed are not independent [ $\chi^2(15) = 26.563, p = 0.03$ ]. The number of active friends on Facebook is not independent to the participants' belief that their friends would be great movie/music recommenders [ $\chi^2(8) = 17.901, p = 0.02$ ]. It was also generally established that both the use of Facebook to find posts about movies/music and the self-proclaimed level of informedness about friends' interests in movies/music are dependent on the percentage of movies/music-related posts [ $\chi^2(3) = 8.345, p = 0.03$  and  $\chi^2(6) = 15.453, p = 0.01$ , respectively]. It was also reported that the users that consider their News Feed posts for music/movies as recommendations from friends, do use Facebook to find such posts [ $t(36) = 3.28, p = 0.002$ ] and also express their interest in such posts [ $t(36) = 2.877, p < 0.01$ ]. The interest itself was manifested in the perceived specific knowledge of actual interest overlap between the users and their Facebook friends [ $F(2,35) = 4.826, p = 0.01$ ]. Additionally, it was evident that the participants thought that the interestingness of friends' posts was the major factor for them to consider Facebook as a premier source of information for movie and music content [ $F(2,35) = 3.363, p = 0.02$ ]. From the above findings, it is evident that there is need for functionality that allows users to identify interesting content based on recommendations derived from the friend's relevance on common criteria (in this case content *genres*) and not based on the number of overall likes or strength of the social connection.



### 3.3 Interaction

The participants reported that their level of awareness regarding their friends' interest in movies and music is directly dependent to the frequency with which they check their News Feed, which was also proportional to the time they spend each day in the network [ $\chi^2(12) = 16.158$ ,  $p < 0.01$ ]. There is significance in the fact that the same users proportionally stated that they thought that at least some of their friends would be great recommenders for movies and music [ $t(36) = 2.754$ ,  $p < 0.01$ ]. As a remedy for the frequency of the required checking of the News Feed, the participants reported that the most helpful requirements for a recommender system would be the ability to view ranked recommendations for movies/music [ $\chi^2(3) = 9.491$ ,  $p = 0.02$ ] and also to edit and fine-tune the recommendations for improved accuracy [ $\chi^2(3) = 8.73$ ,  $p = 0.03$ ]. The later ability is also very important to participants that check their News Feed infrequently ( $z = 3.089$ ,  $p < 0.001$ ). Additionally, another positive correlation exists between the importance of the requirement to view ranked recommendations within Facebook and the general knowledge of friends' interests in movies/music ( $z = 3.035$ ,  $p < 0.01$ ), illustrating the expected positive impact of the particular functionality within a social network.

### 3.4 Expectations for the proposed solution

Users that do not currently consider or use Facebook for finding posts about music and/or movies reported that a more fine-grained processing of friends' posts would result in more interesting recommendations, while they consider the option to easily view a collection of recommendations as very useful ( $t(36) = 2.782$ ,  $p < 0.01$ ). Interestingly enough, for all users, a positive correlation exists between strongly believing (assigning 5 in a 1–5 Likert scale) that a more fine-grained processing of friends' posts would reveal the most interesting recommendations and strongly stressing (again assigning 5) the usefulness of easily viewing a recommendation collection, i.e., filtering out the noise from unrelated posts ( $z = 2.443$ ,  $p < 0.01$ ). Finally, to verify the correlation of the findings, we calculated the expected dependences between the requested fine-grained processing of friends' posts and the frequency that users check their News Feed [ $\chi^2(12) = 28.837$ ,  $p < 0.01$ ], as well as the time (duration) spent on the News Feed [ $\chi^2(12) = 16.158$ ,  $p < 0.01$ ].

Regarding specific expectations, the participants expected overlapping interests with a relative sub-group of Facebook friends and stressed the importance of choosing to extract the information from those specific subgroups to result in accurate recommendations [ $F(2,35) = 3.606$ ,  $p = 0.02$ ]. The ability to view ranked recommendations from friends that have highly relevant interests was a major requirement. This

finding aligned with the expectation that if items of interest were common among friends. That would also lead to certain posts being of interest, and that correlation should be leveraged by the recommendation engine [ $F(2,35) = 4.034$ ,  $p < 0.01$ ]. Taking all the above into consideration, our findings provide the necessary motivation and also set the following generic requirements for the design of the proof-of-concept prototype:

- Allow users access to items from their friends, enabling them to filter the results for selected periods of time.
- Allow users to leverage their existing knowledge of their social circle and easily modify the configurations (i.e., weights) that drive the content selection process, to better reflect interest overlap.

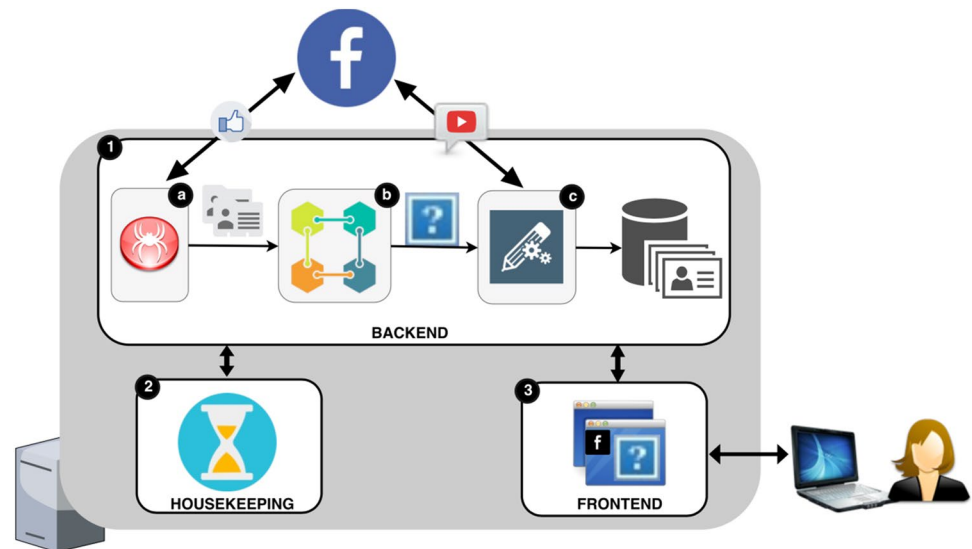
## 4 Methodology and system architecture

By design, the system processes all content published by the user's friends (e.g., posts, links), identifies content belonging to different categories (e.g., music, movies) and then collects information to assign that content to a more specific subcategory [e.g., music (sub)genres]. Subsequently, the system analyzes any text accompanying that content, so as to identify the sentiment regarding that content and infer a positive or negative opinion expressed by the content's publisher. Based on the user's interest profile, and the similarity score with the content's publisher, the previous information is used to assign an interest score to the content that will determine whether the content should be suggested to the user or not.

Previous works focused on people and tags to recommend content to users (Guy et al. 2010). Similarly, Facebook has also implemented a recommender system that suggests content liked by friends (Facebook 2019). Our approach follows a different path; it aims to filter through the massive amount of content that is posted by a user's friends and select the most likely to be of interest to the user. Another key concept behind our system, that differentiates it from other recommender systems, is that it enables users to leverage their existing knowledge regarding the overlapping interests they have with their online friends, in a fine-grained manner. While certain users may have very similar tastes in a specific subcategory, they might have completely different interests in other categories. For example, Alice and Bob may like the same arthouse films, but completely disagree when it comes to romantic comedies.

The proposed method works as follows. First, we collect all the interests found in the user's profile. Every element is processed for extracting information for its categorization (e.g., movie). Next, based on the category, various online services (e.g., Wikipedia) and search engines are employed to collect more information that will allow

**Fig. 1** Overview of main components of the prototype



further categorization (e.g., romantic comedy). After all the elements are processed, an interest profile is created for the user, containing the various sub-categories that describe the user's interests, and a score reflecting their importance. Subsequently, the same process is repeated for each of the users' friends, and a similarity profile is created for each friend. This profile contains the overlapping interests of the two users, i.e., the categories and sub-categories that are of interest to both users. Depending on the number of overlapping elements (e.g., both users like the films "Fight Club" and "The Matrix") a similarity score is assigned to each sub-category. After the initial processing, the user can manually "tweak" the similarity score for each friend. A low score represents a difference in taste for the specific category. Furthermore, the system monitors which content the user clicks on to dynamically update the similarity scores for the specific content's publisher. A feedback rating function allows the user to rate recommended content, which also updates the publisher's similarity score.

It is important to note that our system is not intended as a replacement for existing content selection algorithms for users' News Feeds; instead, we aim to identify the most interesting items (e.g., YouTube videos) posted by a user's friends, which may be otherwise lost amidst the massive amount of generated content. These selected posts are to be presented in a separate media recommendations section, for easy accessibility, and to adhere to Facebook's menu/resource design, each dedicated to a specific category, thus expanding the existing functionality of the social network and improving the user experience.

The system consists of three main components, as shown in Fig. 1:

1. *Backend* The main component of the system. The functionality can be broken down to three tasks:

- (a) *Bootstrapping*, where every *like* under categories of interest in the user's and her friends' profiles are gathered and analyzed, for extracting entities and conducting a fine-grained categorization. These data are then leveraged for creating each user's *interest profile*.
- (b) *Similarity calculation and score tweaking*, in which the system compares the user's *interest profile* to that of her friends and assigning a *similarity score* to each one. At this stage, the user can manually tweak any *similarity scores* to her liking, thus modifying the weights for specific sub-categories or friends. Essentially, this step builds the required social knowledge which will enable our system to later accurately identify content that matches the user's interests.
- (c) *Post analysis and list generation*, where our system processes the text-accompanying content published by the user's friends, using techniques such as sentiment analysis and named entity recognition. This step, basically, aims to extract semantic information from posts and to infer whether the poster has expressed a positive or negative opinion about the media content. This allows the system to assert whether it overlaps with the user's categories of interest, to calculate its significance and, ultimately, to decide whether it should be recommended to the user. To reduce the latency introduced by accessing large collections of online data, we created an offline knowledge base con-

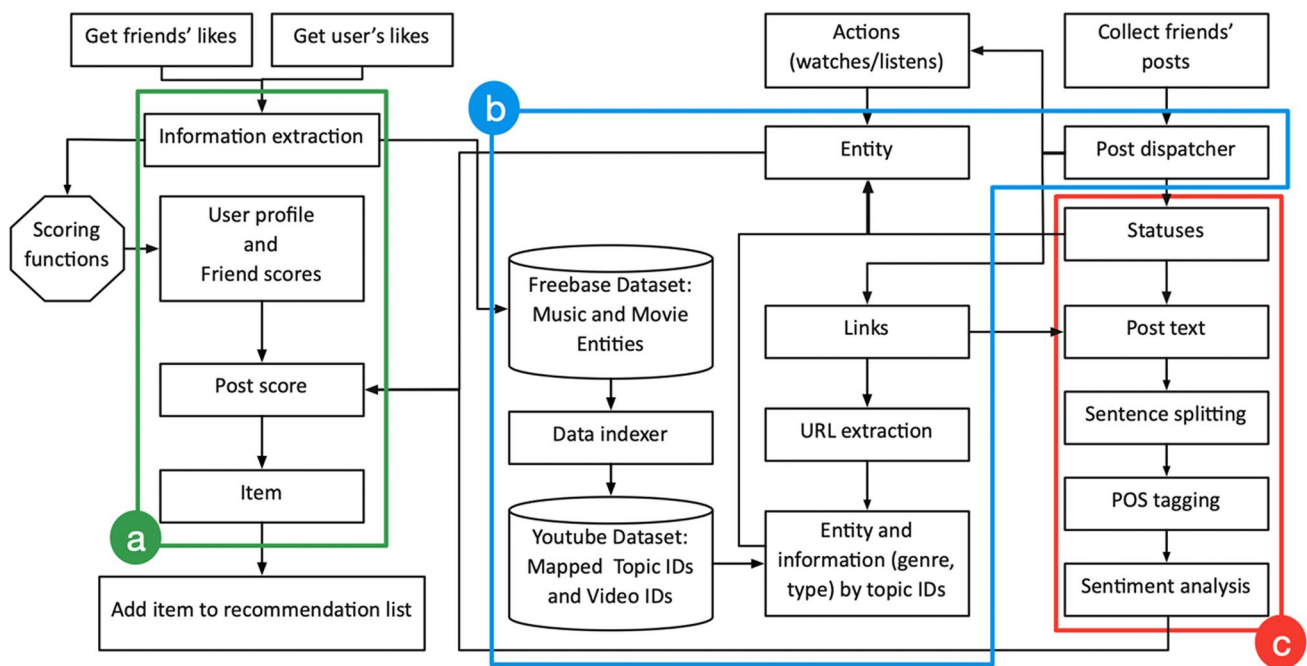


Fig. 2 Detailed workflow of the processes and tasks

taining entities that are of interest. The knowledge base basically acts as a data cache to avoid excessive network traffic and, thus, operational overhead delays. Our *entity extraction* mechanism relies on the knowledge base to swiftly extract entities, but also leverages online resources for data not found in the knowledge base.

2. *Housekeeping* This component is responsible for *housekeeping* and consists of two distinct modules with separate functionalities, which are executed periodically. One module refreshes the recommendation lists for each user so as to contain fresh content. The other module, which is executed less frequently, polls users' profiles to account for any changes made to their likes and interests. This is done for recalculating necessary scores and weights based on new information.
3. *Frontend* This is the interactive component of our system, where users can view the lists of recommended content or tweak the weight of their friends to better reflect the desired similarity score for specific (sub)genres.

The backend tasks: (a) bootstrapping, (b) similarity calculation and score tweaking and (c) post analysis and list generation, consist of several processes that are interconnected; Fig. 2 depicts the full process and task workflow of the methodology.

In the following paragraphs, we analyze the aforementioned components and tasks, further.

Table 1 Notation used for the bootstrapping phase

	Variable	Description
GS	GenreScore	Genre occurrences in user's/friend's set of likes
GSL	GenreScoreList	List of genre scores for a user/friend
OGS	OverallGenreScore	Sum of genre scores in GSL
LS	LikeScore	Avg of GS for genres associated with a like
LSL	LikeScoreList	List of LS entries for a user/friend
OLS	OverallLikeScore	Sum of like scores in LS

### 4.1 Bootstrapping

This component is responsible for the entire bootstrapping phase, where the required data about the active user and her friends (hereby referred to as a clique) are collected and processed. We create interest profiles from *likes* found on their personal Facebook profiles, namely, under the music and movies endpoints (Facebook 2017). Furthermore, in order to provide fine-grained recommendations, accurate and detailed genre identification for each *like* is crucial. Consulting our knowledge base, we manage to extract entities and information about their genres<sup>1</sup> and categories. Table 1 provides an easy reference for our notations used in the

<sup>1</sup> For simplicity, we use the term "genre" to also denote fine-grained sub-genre information.

**Table 2** Sample *genre* score list

Genre	Genre score	Liked page
Rock music	2	The rolling stones, AC/DC
Rhythm and Blues	1	The rolling stones
Country	1	The rolling stones
Pop music	1	The rolling stones
Hard rock	1	AC/DC
Heavy metal	1	AC/DC
Alternative rock	1	P!nk
Pop punk	1	P!nk
Dance-pop	1	P!nk

bootstrapping phase, which are analyzed in the following subsections.

#### 4.1.1 Genre and like scoring (GS and LS)

Facebook allows users to express their interests through *likes* associated with specific objects such as pages and posts. However, not all *likes* are equally representative of the user's preferences and identifying which should have more contribution during the content selection process is critical for providing accurate recommendations. By analyzing all of the user's *likes*, we calculate the significance of all identified *genres* for the user. Each *like* represents a specific entity which corresponds to a set of associated *genres*, as specified by our knowledge base.

#### 4.1.2 Genre and like scoring lists (GSL and LSL)

We create a GSL for every user in the clique, a key-value table where each *genre* is the key and the number of occurrences throughout all *likes* is the value. An example GSL for a user with 3 *likes* in her Facebook profile is shown in Table 2.

#### 4.1.3 Overall genre and like scores (OGS and OLS)

In order to normalize the *genre* similarity between the user and her friends at a later stage, we also calculate an *overall score of genres* (OGS) for each of them as shown in Eq. 1:

$$\text{OGS} = \sum_{i \in \text{GSL}} \text{GS}(i) \quad (1)$$

where  $\text{GS}(i)$  is the *genre* score for a specific entry in GSL. In addition, we create a LSL for every clique member's *likes*, to assign weights using the GS values in the GSL calculated above. LSL is an average calculation of the GS corresponding to the *genres* found both in the *like* item and in the GSL. For example, "The Rolling Stones" get a 1.25 score as the

associated *genres* are "Rock Music," "Rhythm and Blues," "Country" and "Pop Music." As shown in Table 2, they have 2, 1, 1, 1 scores for the *genres*, thus  $5/4 = 1.25$ . We chose to use such metric in order to weigh *likes* according to the *genre* preferences of the user. The LS scores in the LSL are used for calculating similarity with other users. We also create an *overall like score* (OLS):

$$\text{OLS} = \sum_{i \in \text{LSL}} \text{LS}(i) \quad (2)$$

used later for normalization when calculating *like* similarity. As a fallback option, if there is not enough information for a specific user, we crawl through their post history and attempt to identify and extract entities for creating their *interest profile*.

## 4.2 Sentiment analysis

Sentiment analysis is the process of mining, extraction and identification of the subjective information in texts. For this work, we utilize the positive and negative sentiment of the sentences to filter out the negative polarity texts. Sentiment analysis is conducted before any post is analyzed for named entity recognition. If the message contained in the post has negative polarity, it is discarded (as our goal is to recommend content that was endorsed by the user's contacts). Only when the post's message is classified as positive, it is retained for further analysis. We assume that the absence of text in a specific post implies a positive polarity. In the following subsections we analyze the components and tasks, used in our proposed system, concerning the *sentiment analysis*.

### 4.2.1 Analyzer

A crucial aspect of recommendations, especially when natural language is involved, is detecting the reviewer's point of view on the topics and subjects they post; for instance, the text that accompanies links shared on Facebook. Sentiment analysis defines the method of discerning the positive feeling (attractiveness) or negative feeling (aversiveness) in a text. A large body of research has explored this subject, from detecting sentiment in literature texts (Nazir et al. 2019), to classifying emotion and sentiment in social media and microblogs (Dridi and Reforgiato Recupero 2019; Antonakaki et al. (2017)) as well as the polarity in product reviews (Fang and Zhan 2015). For the proposed approach, we need the ability to discern the poster's opinion, for making correct recommendation decisions. Specifically, we use a modified version of the Semantic Orientation CALculator (SO-CAL) approach introduced by Taboada et al. (2011). SO-CAL is used for extracting sentiment polarity and strength from text



and consists of the proposed algorithm and a set of dictionaries categorized by their part of speech. There are 6 different dictionaries in the set containing 1542 nouns, 1142 verbs, 2824 adjectives, 876 adverbs and 217 valence shifters. We opted to use SO-CAL due to its ranked dictionaries with words scored with sentiment intensity (valence) for fine-grained sentiment tagging and also the scoring heuristics which perform better than other dictionary-based approaches we tested. Additionally, we made some modifications, such as changing the semantic orientation (SO) value of certain words, simplifying the negation lexicon and adding an emoticon lexicon which we created manually using info from Wikipedia, which contains 110 entries.

#### 4.2.2 Text preprocessing

To correctly analyze text, we first filter out any Facebook tags (with @) and non-English or non-printable characters. We use the NLTK toolkit (Bird et al. 2009) and the PyCLIPS module for text processing (Garosi 2008). Specifically, we leverage PyCLIPS for text splitting and tokenization, while NLTK is used for part of speech (POS) tagging as it proved to be more accurate.

#### 4.2.3 Semantic strength tagging using dictionaries

The SO-CAL uses dictionaries with words grouped by their POS, ranked with valence strength  $[-5,5]$  (inclusive, 0 is excluded) (Polanyi and Zaenen 2006). POS tagging is necessary as a word may be defined with different POS value, which results in different valence strength (Taboada et al. 2011). For example, the word “best” has a SO value of 5 as an adjective, while it has 0 (neutral) as a verb. Each word in our defined sentence gets assigned a value depending on its POS tag. Additionally, as the use of emoticons is widespread in social media, we handle them accordingly. If an emoticon is found in a post, we check whether it exists in our corpus; if it does, it is handled as any other normally ranked word. For example, “:)” is ranked with a score of 3, whereas “:(” with  $-4$ .

#### 4.2.4 Valence shifters

Valence shifters are words that carry different semantic values than the words described so far, and their POS tag does not necessarily affect their use. They are called shifters as they change the strength or effect of a lexical item when they are nearby (Polanyi and Zaenen 2006) in a sentence. Their area of effect is limited, which is defined by various grammatical properties. Each valence shifter is assigned a SO value, although it is applied differently on the score calculation. Specifically, it works as an additive multiplier on

the initial SO value of the lexical item it shifts. The default multiplier for every word is 1. For instance, “robust” has a SO value of 2, and “really” has  $+0.2$  multiplier which consequently is an intensifier; thus, “really robust” is assigned a score of  $(1 + 0.2) * 2 = 2.4$ .

#### 4.2.5 Negation

There are many approaches to apply semantic negation. One of them is switch negation (Sauri 2008), where the SO value of a lexical item is reversed; e.g., “good” has a SO value of 1, “not good” gets a SO value of  $-1$ . Horn (1989) proposes that there is no semantic symmetry between negative and affirmative sentences. Supporting that, Taboada et al. (2011) state that applying shift negation is more a realistic approach linguistics-wise. This way, instead of reversing polarity, the SO value of a term is shifted toward the opposite polarity by a fixed amount. In our example above, when following the SO-CAL exact directives, the value is 4. As a result, while a term with a strong meaning like “best” has a SO value of 5, the phrase “not the best” will result in a SO value of  $-4 + 5 = 1$  instead of  $-5$ . Similarly, the word “terrifying” ( $-3$ ) in “not terrifying” gets a SO value of 1. In our case, negators are basically of two types; the word “not” and any word containing the suffix “n’t,” like “couldn’t” or “shouldn’t.”

#### 4.2.6 Scoring

As explained above, valence shifters and negators are applied as modifiers to the SO-CAL value of the lexical term they refer to. To calculate the final SO value of a lexical term, we recursively apply any modifier value found searching backward, until a determiner (e.g., a comma or sentence connective) is found. For example, “expensive” has a SO value of  $-1$ , and the phrase “very expensive” has a SO value of  $(1 + 0.2) * -1 = -1.2$ , and “not very expensive” applied recursively will be  $(-4) * (-1) + [(1 + 0.2) * -1] = 2.8$ . This calculation is applied to every lexical term in a sentence, and finally, the sum of every sentence’s SO value is aggregated, producing the total SO value of a given text. After manual analysis of a corpus of posts, we set a threshold of 0.7 for the overall SO score, after which we consider the text to be positive.

#### 4.2.7 Irrealis blocking

Irrealis blocking, as described in the SOCAL documentation and previous work (Taboada et al. 2011; Liu 2015), is used to describe situations where something has not happened yet as the speaker is talking and, thus, the result or action is uncertain. This contains subjunctive, conditional and imperative moods which can be detected by a pattern

module. Some of them, such as the imperative mood, may be semantically significant in our case, as it can be used to express sentiment upon a subject. That means, words in the effective radius of an irrealis marker (such as modals and quoted sentences) have a nullified SO value. In addition, we ignore any sentences that are grammatically categorized as questions.

#### 4.2.8 Slang and typos

Slang terminology can impact the accuracy of our content processing. Similarly, abbreviated versions or mistyped words are common instances, and further exploration is required for effectively handling all such cases. However, major services such as Facebook are already focusing on processing and automatically understanding slang terminology (Wired 2013). We made some initial exploratory experiments with the Urban Dictionary (2019), in an attempt to handle such instances. Specifically, we crawled the Urban Dictionary (2019) and extracted roughly 7 million entries along with their definitions. We used the definitions to decide whether a word had positive or negative semantic meaning, depending on the number of times a definition was labeled by our sentiment analysis module. However, as there are no restrictions enforced in the urban dictionary, a large number of words we acquired contained a significant semantic noise. The shortcomings of following an automated procedure to make dictionaries for that purpose, as well as increasing the size of dictionaries, have been previously documented by others (Taboada et al. 2011). Our initial experiments with integration of the urban dictionary resulted in a significantly reduced accuracy. A suitable approach is to manually classify any word used for such a task, but this is too time-consuming to be applied to 7 million words. A manual selection of a significantly smaller subset could offer a solution. As SO-CAL works with part of speech tags, any new addition needs to be tagged. However, the size of our urban dictionaries, uncommon words (that are hard or impossible to be identified by taggers) as well as the fact that it contains phrases, further complicated this task. Due to these obstacles, we ruled out the use of the dictionaries we made from urban dictionary as their use decreased our analyzer's performance.

### 4.3 Entity extraction

In order to gather the needed information about the friends' posts, we have to analyze their entire activity. Our purpose is to define all the entities in a friend's post and, subsequently, keep those that belong to categories of interest. The system utilizes three endpoints (Facebook edges) from the Facebook Graph API to obtain the needed data, Links which are posts

containing a URL and two kinds of *actions* which are posts with user-generated social stories (Facebook 2018). Specifically, we use the listening and watching social stories.

We also leverage the Freebase (2018) collection for creating a knowledge base of entities. Google's Knowledge Graph Search API was partly powered by Freebase, which was recently deprecated and replaced by the Graph API. We extracted every entity under musical artists/bands and movies coupled with detailed *genre* information, through their API. Specifically, we obtained 293,506 unique entities, 221,091 movie entities and 72,415 musical artist/bands entities paired with *genres*, as JSON objects. Each entity in the knowledge base carries a unique topic ID as provided by Freebase that we use for entity matching with YouTube videos. We developed a dictionary-based entity extractor, which utilizes resources from our knowledge base. In addition, to swiftly map a video to its respective topic (entity) we developed an indexer that maps each topic ID to its entity and every corresponding video ID found in YouTube at that time. This resource serves as a data cache that enables fast named entity recognition, without relying on online APIs, and also reduces excessive network traffic and delays due to operational overhead. In the following subsections we analyze the tasks, used in our proposed system, concerning the *entity extraction*.

#### 4.3.1 Link analysis

Facebook defines as *Links* any posts containing a link/URL posted alongside a message. To extract entities from them, we take advantage of Facebook's embedded URLs in posts, which provide information about the shared link. Our system specifically searches for YouTube URLs, using a video ID-to-topic ID pair matching. From every URL, we extract the video ID and find its corresponding pair of topic ID in the knowledge base. The entities in the knowledge base are cached and indexed with their unique topic ID, enabling us to directly extract any of them related to a given YouTube video. Figure 3 presents the workflow of our link analysis.

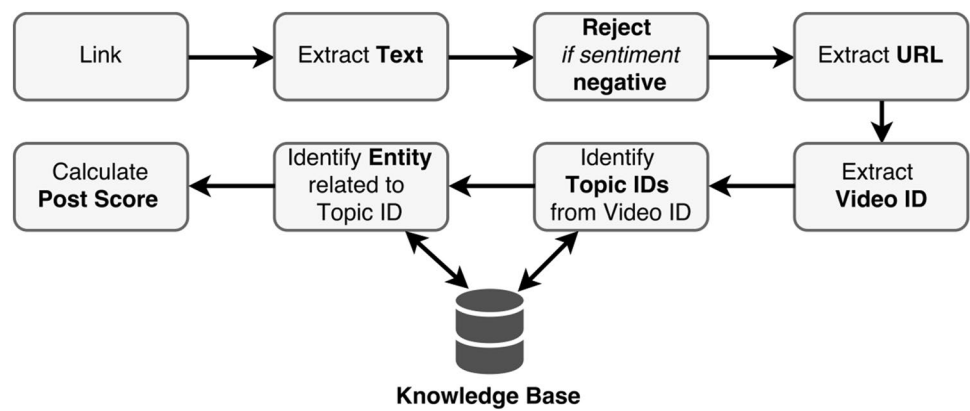
#### 4.3.2 Actions

Entity recognition in these posts is straightforward, as the name or title of the item is found within the post's data. Facebook prohibits fetching comments accompanying *actions*. However, for systems that are designed to be deployed by Facebook, in practice, they would also have access to the comments.

### 4.4 Quantifying similarity

A challenging aspect of the content recommendation process is selecting the similarity formula that will quantify

**Fig. 3** Workflow of the link analysis process



how interesting a specific post will be to the user. In previous work (Gretarsson et al. 2010) the proposed system used an interactive graph, where item and friend weighting were manual and required the user’s interaction even in the initial stage. Scores were also in a capped scale (1–5 inclusive), thus, not being as fine-grained as needed. To that end, we opted to employ the Jaccard coefficient and created a formula that is well suited for automatic similarity calculation in sets with weighted items and of arbitrary size. Specifically, we devise a formula that contains certain modifications to the Jaccard coefficient, as we describe next. To calculate the similarity among a user and her friends in a fine-grained manner, we leverage the *genre* and *like* scores calculated in the bootstrap phase; our in-depth profiling of each user’s interests enables us to develop an accurate mechanism for “scoring” friends and posts. Table 3 lists the variables used for the similarity calculations.

To calculate *genre* similarity score between a user and a friend, we aggregate the sum of scores of each overlapping *genre* and then divide by the sum of their OGS to normalize.

Using formulas (3) and (4), we calculate the GSS and the LSS that the (active) user has with a friend in either category; these two scores are combined using formula (5) into the FSS, which represents the actual overlap between the two users and is used for post score calculation in a later stage.

$$GSS(U, F) = \sum_{i \in G_{SL}(U) \cup G_{SL}(F)} \frac{GS_U(i) + GS_F(i)}{OGS_U + OGS_F} \quad (3)$$

The same principle applies for the LSL table:

$$LSS(U, F) = \sum_{i \in L_{SL}(U) \cup L_{SL}(F)} \frac{LS_U(i) + LS_F(i)}{OLS_U + OLS_F} \quad (4)$$

Based on our initial observations, we realized that the applying weight to the equation would increase the accuracy and the precision of our results. Therefore, we experimented by applying different weights while running predefined use cases, which lead us to the conclusion. We find, by tweaking

the weights in Eq. 5 after manual testing, that a weighted average is suitable, giving *genre* overlap double the weight (content classification proved to be a more defining factor than the *likes*), as it proved to boost accuracy in a more fine-grained fashion.

$$FSS(U, F) = \frac{2 * GSS(U, F) + LSS(U, F)}{3} \quad (5)$$

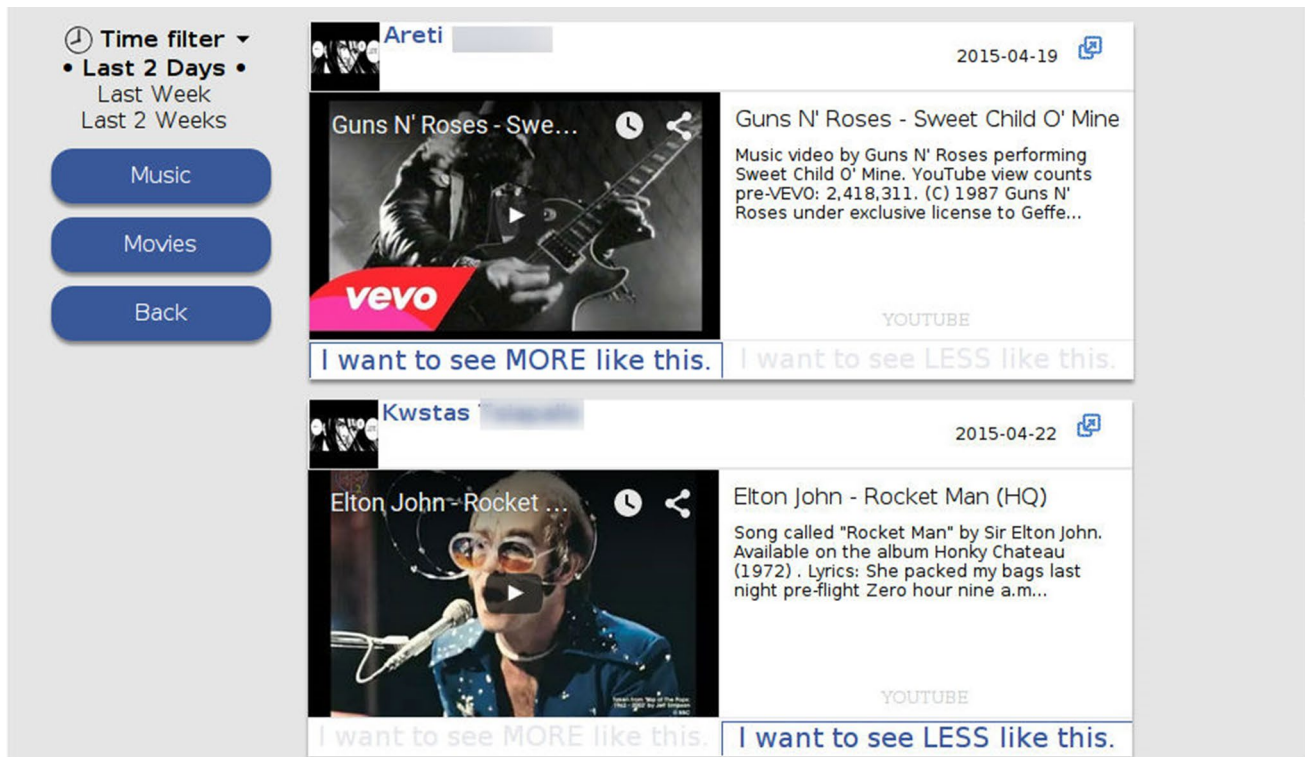
Finally, in order to populate the recommendation lists, we gather all the posts contained in the user’s friends’ profiles and apply the formula shown in Eq. 6. Each *PS* result denotes how interested a user would be in that post, based on that user’s *genre* scores that overlap with the post’s *genres* and how similar preferences the user and the poster have.

$$PS = \frac{\sum_{i \in G_{SL_U}(PG)} GS_U(i)}{|G_{SL_U}(PG)|} * FSS_{poster} \quad (6)$$

The resulting item score is not capped, as the values of *genre* and *like* scores cannot be predicted beforehand. This approach also provides a discreet ranking among items. The intuition behind multiplying with the FSS is to avoid the aforementioned instances where users are shown content irrelevant to their interests.

**Table 3** Notation used for similarity calculations

Variable	Description
GSS	GenreSimilarityScore <i>Genre</i> similarity between user and friend
LSS	LikeSimilarityScore <i>Like</i> similarity between user and friend
FSS	FriendSimilarityScore GSS/LSS similarity between user and friend
PS	PostScore Rank of a friend’s post
PG	PostGenres <i>Genres</i> extracted from a post



**Fig. 4** The recommendation page of our proof-of-concept implementation (frontend in Facebook)

## 4.5 Facebook application

In this section, we provide details of our prototype implementation of the system *frontend*. We implemented it as a Facebook Canvas app that leverages the Facebook Graph API. In the following subsections we analyze the tasks of *initialization and score adjustment* and *top recommendations*.

### 4.5.1 Initialization and score adjustment

Upon installation, the system initially creates the user's *interest profile*. Once the profile is created, the user is taken to the score adjustment page. It contains a list of friends and similarity scores for each category and *genre*. Contacts with no overlapping interests in any category are omitted. The user can tweak the scores for each category and *genre* on a scale of 0–100. This part of our application is presented at first during the initialization step and can be accessed at any time by the user for further tweaking the scores.

### 4.5.2 Top recommendations

As can be seen in the screenshot shown in Fig. 4, our prototype allows users to navigate to the respective recommendation lists for movies and music and can also change the granularity of the time window from which the recommendations

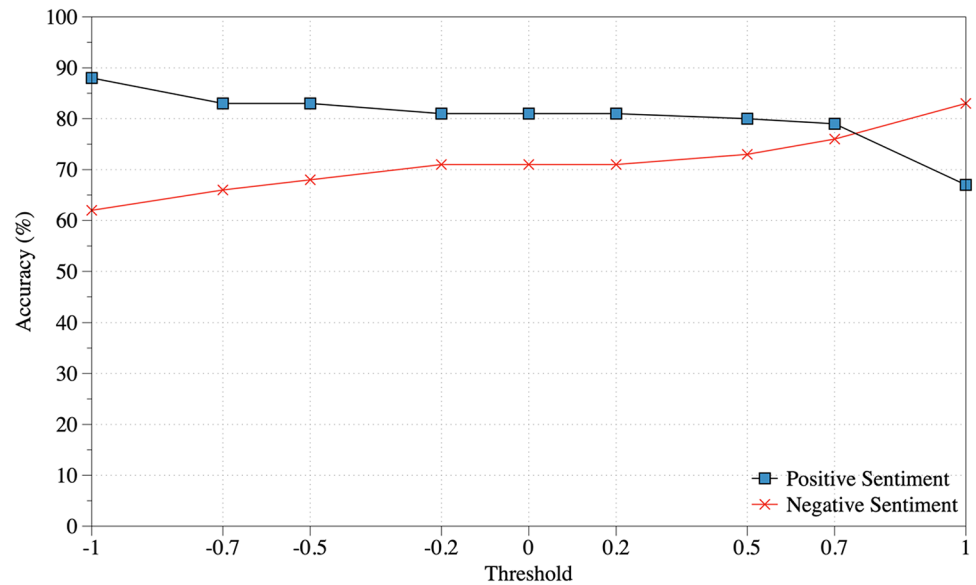
are shown. Our prototype allows users to increase that window to include up to the past 2 weeks and present up to 10 recommended items, but this can be trivially changed to allow arbitrarily large numbers. Furthermore, if a list is empty, we obtain the most important *genres* from the user's *interest profile* (after searching YouTube for the user's three highest ranked *genres*) and recommend three random videos from the current category. Each item in the list contains all the necessary information for the users to access; the friend that posted the content originally, publishing dates and the embedded video related to the entity mentioned in the friend's post, along with the video's description as provided by YouTube. Users can also view the original post by clicking on the pop-up icon. Furthermore, users are able to express their opinion on the recommended items. Pressing one of two buttons located at the lower boundary of each item, will tweak the importance of *genres* that the post fits within the user's *interest profile*.

## 5 Evaluation

In this section, we report on our experiments aiming to evaluate:



Fig. 5 Threshold analysis



1. the effectiveness of the heuristics and algorithms for the proposed recommendation method and the proof-of-concept implementation and
2. the user satisfaction regarding the offered recommendations over a long period of time, as well as the generic usability of the proposed method.

## 5.1 Effectiveness of heuristics and algorithms

The first set of experiments was aimed at evaluating the effectiveness of the proposed heuristics and algorithms. For each experiment, we describe the preparation, the datasets used and, finally, present our findings. A critical phase of the recommendation process is the entity recognition, which allows the identification of the entities that a specific post is related to and the definition of the genres that are associated with it. To that end, we measured the effectiveness of our module that maps the unique video IDs to unique topic and entity IDs. We created a dataset by crawling 20 popular music/movie Facebook group pages. In order to ensure accurate results, we filtered out the posts that did not contain a valid YouTube URL; our final dataset contains 5310 links to YouTube. Our module was able to identify 4743 valid title entities, achieving a coverage of 89.32%, demonstrating the robustness of our approach.

### 5.1.1 Sentiment analysis threshold

Our next step was to evaluate our sentiment analysis module and its effectiveness in providing accurate results regarding the sentiment of a given post's content. We obtained datasets published in previous studies that contained English Tweets and Facebook comments, rendering them a suitable sample

of the type of content we expect our system to handle. Specifically, we used the following labeled datasets as *ground truth* for evaluating our approach:

- Twitter Dataset (Sanders 2011) (5513 tweets)
- Facebook Comments Dataset (Zhang et al. 2015) (1000 comments)

We preprocessed the datasets to remove text with a truly neutral sentiment (text with a sentiment score of 0), as such text will not offer any valuable semantic information about the published content. Then, to regulate the semantic noise (i.e., false positives/negatives) we experimented with different sentiment score thresholds, which specified whether a post is classified as positive or negative. The outcome of our algorithm for each text was compared to the polarity label in the aforementioned corpora, to calculate the accuracy. The best results were achieved for a threshold of 0.7, by correctly labeling 79% and 76% of the positive and negative samples, respectively (Fig. 5).

### 5.1.2 Relevance of content

The data were collected from 38 test subjects that installed the Facebook application. We focused on the posts that belong to one of the following three Graph API Edges: *Links* (any post with an embedded YouTube video), *Watches* and *Listens*. Table 4 depicts the statistics for relevant content identified by our system's heuristics, extracted from 3493 accounts that were connected to our participants. Interestingly, we identified many cases of *Links* with invalid URLs; this included old YouTube links that were no longer available or that were malformed (most likely due to users' copy pasting only part of the link to their post). As a result,



**Table 4** Dataset break down for posts with relevant content

Type	Number of posts
Total	521,685
Filtered out (irrelevant, broken links)	277,356
Processed	244,329
Relevant	75,694
Links	45,438
Watches	28,994
Listens	1262
Movies	28,994
Music	46,700

59.98% (94,626 of the 157,762) of the link posts were broken and, thus, removed. Also posts under categories outside of music and movies (i.e., statuses, photos, etc.) amounted to more than 50% of the initial dataset and filtered out as well. Of the remaining valid posts, 30.9% contained relevant content under the music and movies categories, indicating that there is an abundance of content being published that could lead to interesting content being overlooked by users or ignored by the current selection algorithm of Facebook’s News Feed.

**5.1.3 Fine-grained versus standard (Facebook)**

In order to explore whether the users missed relevant content due to Facebook’s standard personalization algorithm, we gathered every News Feed post from  $N=38$  participants (age  $M=28.3$ ,  $SD=3.12$ ; 68.4% male), with number of friends  $M=612.7$ ,  $SD=191$ , that gave us access to their data for a time period of 2 weeks (maximum allowed by the Graph API). We compared them to the content posted by their friends and also processed the data with our approach to

**Table 5** FB News Feed items versus fine-grained items

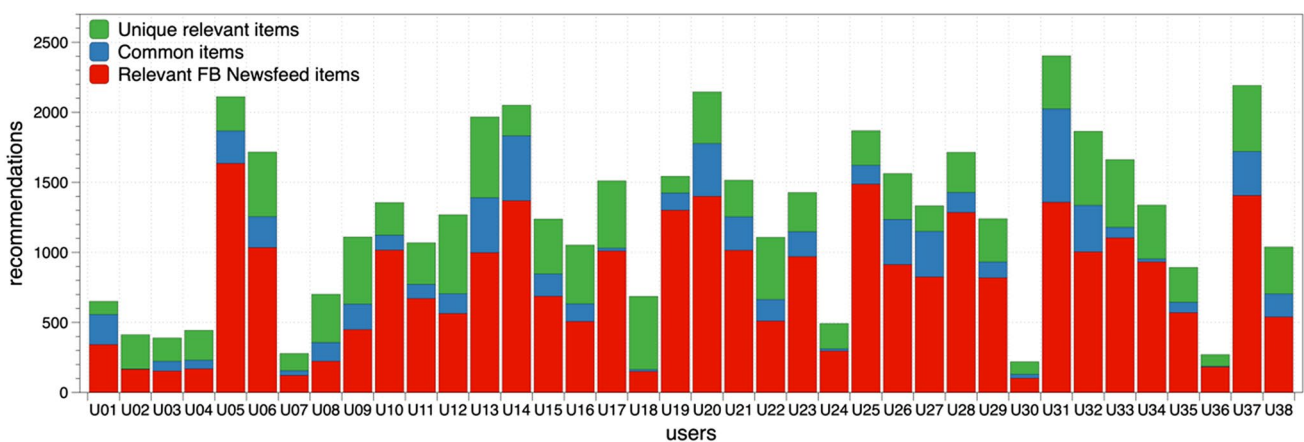
FB News Feed items	Relevant FB News Feed items	Common items	Unique recommended items
94,692	29,282	6487	12,096

identify posts of interest. We then compared the two datasets and identified the common items (Fig. 6). The users’ News Feeds contained a total of 94,692 items, while our system selected 6487 posts that were also present in the News Feed, but also revealed 12,096 additional posts that are suitable for recommendation, as they match the users’ *interest profiles* (Table 5).

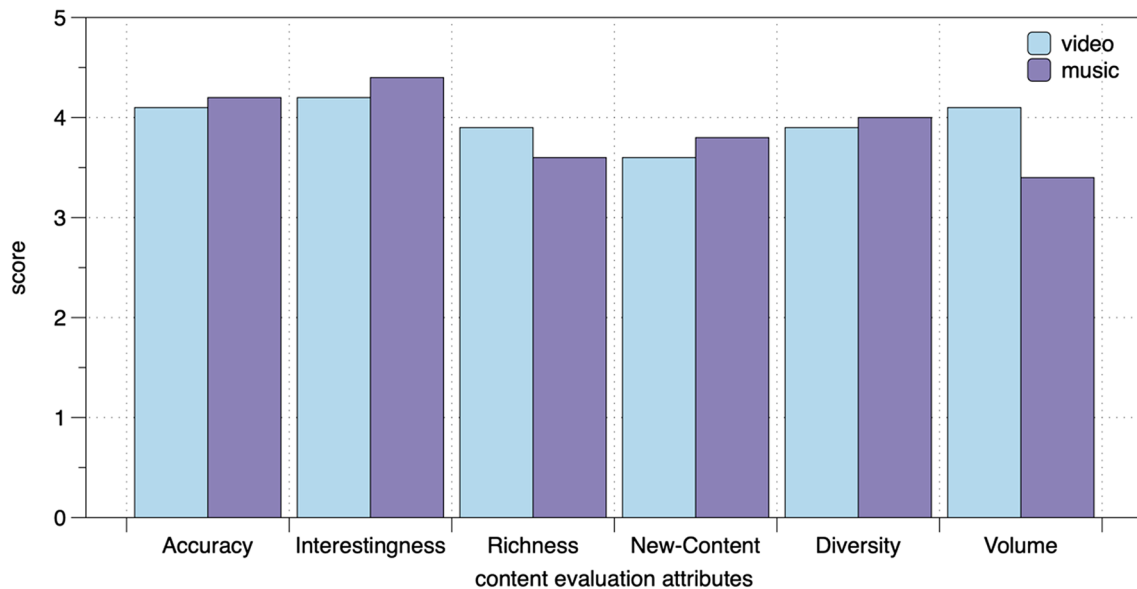
Out of the full content in the News Feed, it was identified that 30.9% was relevant to the categories of interest. From the total content in the News Feed, the system identified an additional 12.7% verified unique relevant content (fine-grained recommended relevant items that satisfied the user profiles) plus a 6.8% verified common relevant content (fine-grained recommended relevant items that both FB News Feeds and fine-grained approach had in common). This calculates to 41.3% verified unique relevant content plus 22.2% verified common relevant content out of the total relevant content in the News Feed.

**5.1.4 System runtime evaluation**

We note here that while the OGS and OLS computation is an online process (computing scores for genres and likes, in order to make fine-grained recommendations to users), the computation of the SO values, which incurs an additional processing cost, can be performed in an offline fashion



**Fig. 6** Number of recommendations revealed by our method, compared to the content presented in the users’ news feeds by Facebook



**Fig. 7** User feedback on recommendations (content)

or be offloaded to a different machine. While theoretically the SO values for any user may change whenever a new item enters the database, the magnitude of such a change is small (or even zero, when similar text has been processed already). Hence, in real-world conditions, the recalculation of the SO values can be performed periodically, or when the new content reaches a threshold such as 10% of the existing processed text.

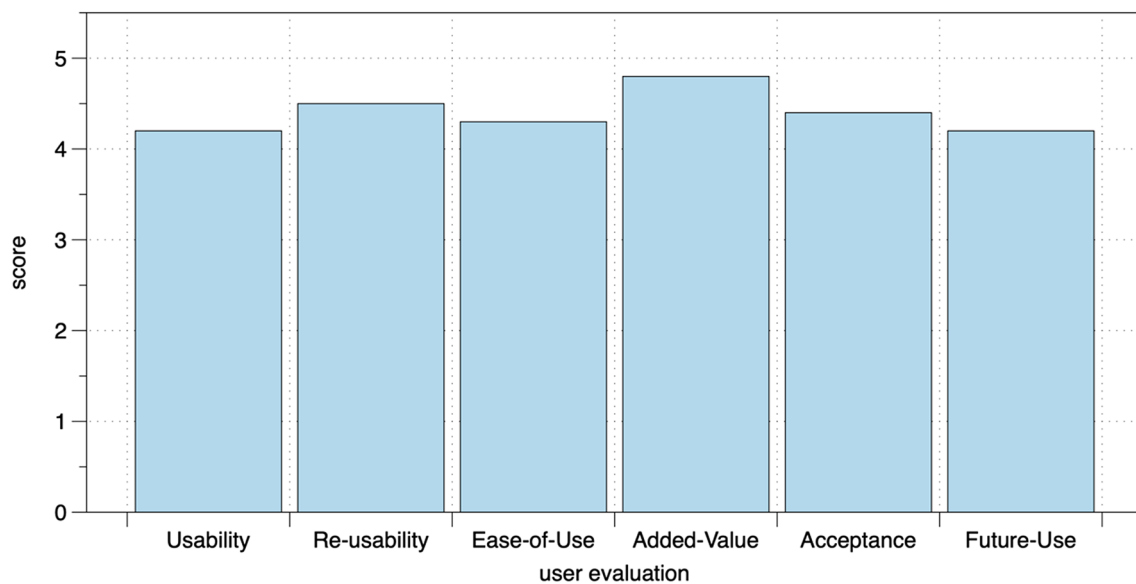
Regarding the user interface for the user feedback on the recommended items, the average number of clicks per user per day was 3.2. The input for the evaluation experimentation (questionnaire response and relevance marking of items for verification purposes) was part of the experiment execution and thus is not included in the calculation. The feedback required by the users was mostly during the first 3 days when the average click count was 7.2, which was performed in less than 1 min using the provided user interface. After that the average click count dropped to an average of 1.9.

## 5.2 User satisfaction

In the next experiment, we conducted a pilot deployment of the prototype application for a preliminary usability study aiming to examine how the users perceive the unique relevant content recommendations over a long period, as well as the generic usability of the proposed method. The participants were recruited anonymously, through institutional mailing lists with two requirement criteria for participation, age group 22–38 (for compliance with the initial user study) and > 200 number of friends to allow enough relevant posts for recommendation. The selected participants  $N=18$  (age  $M=29.0$ ,  $SD=6.4$ ) were not part of the initial user study

group and had adequately high numbers of Facebook friends ( $M=319.7$ ,  $SD=136$ ). The focus was to subjectively evaluate the approach over a longer period of time in order to get more indicative results on long-term user experience. The goal was to allow the participants to extensively interact with the system within Facebook, using their own accounts in a 2-week-long study aimed to evaluate the effectiveness of the approach, and address specific usability and acceptance factors that would enable a larger, more complex, validation. User feedback was collected using an adapted Questionnaire for User Interaction Satisfaction (Chin et al. 1988) on the content of the fine-grained recommendations and the user experience while interacting with the application. The users were asked to install the prototype application and use it for at least a few minutes each time, and at least three times within each week, for 14 consecutive days. They were asked to keep track of any new content the system presents, the level of interest for each recommendation, the accuracy of the scoring and to explain, if used, the need to adjust scores or ranking of the recommendations in a written form think-aloud protocol. The usability feedback was collected via online forms (1–5 Likert's scale) and the full interaction feedback from assigned online diaries. One of the major findings was that the users were given adequate number of recommendations, taking into account the number of Facebook friends they had. From those, on average, 31% were new posts with more than 80% classified as interesting.

As seen in Fig. 7, users reported accuracy and interestingness of video and music recommendations of 82% and 84% on average (values of 4.1 and 4.2 on the Likert scale), respectively. The new content revealed by our prototype was deemed very adequate and quite rich and diverse. Those



**Fig. 8** Usability evaluation

qualities were attributed based on the use and revisiting of the application in the 2-week time span of the evaluation session. This is a direct improvement over the initial user study where casual users reported that it was disappointing to have to spend time and effort to scan their timelines in Facebook for interesting recommendation posts. The recommended video content was larger in volume and has been evaluated as marginally richer than the music content.

Furthermore, the feedback on the system's usability was positive, part of it attributed to the adopted simple and clear design, as shown in Sect. 4.

As can be seen in Fig. 8, the participants easily navigated and used the application for the whole duration of the evaluation. After the 2 weeks of use, the users reported that they were pretty familiar with the way the application works and easily integrated the system into their social activities' workflow. The general belief was that this recommendation approach would be a welcomed addition for other domains that they may require recommendation in the future, recognizing a significant potential in the added value of our proposed approach.

## 6 Conclusions

In this paper we explored the utility of a recommender system within a social ecosystem, designed to identify content published by the user's friends that matches a fine-grained interest profile that is automatically generated from each user's interests reflected by social data (i.e., *likes* and *groups*). Our approach is designed as a complimentary mechanism to the main content selection algorithm already

in place in social networks (e.g., News Feed in Facebook), which is driven by the amount of interaction with each friend. While these are intuitive and effective selection criteria for general content (e.g., general posts and life events), it is not optimal for "entertainment-based" content. Our driving motivation was that content that significantly matches a user's interests can be missed due to weak social ties with the friend that published it.

The qualitative user study allowed us to identify the expectations of participants regarding the ability of their friends to publish content of interest for our supported categories (i.e., music and movies), which guided the subsequent design of the approach. Our proof-of-concept implementation as an app for Facebook allowed us to explore the practical aspects and intricacies of processing and extracting information from user-published content. The user evaluation asserted the effectiveness of our approach and the suitability of online social networks as an information-rich ecosystem for providing fine-grained recommendations to users.

The proposed approach resulted in the recommendation of 41.3% of additional unique content when compared to the standard Facebook recommender. The user evaluation revealed that over 80% of the unique content was deemed as interesting by the participants. Other qualitative aspects, such as newness, richness and diversity, were all rated in the top quartile of the evaluation scale.

In our experiments, we focused on two specific types of content, namely movies and music. However, our approach is applicable to a number of topics and could be extended for supporting various categories. Potential directions could include a wide range of topics including the literature, sports, traveling and politics. This will also require our

system to expand its content analysis process to handle status updates, which can be problematic for the entity extraction step, as described in Sect. 4.3.

A significant expansion of our current approach would be to develop techniques for extracting entities from Facebook status updates, even when no media content is contained. This requires more complicated heuristics and relatedness techniques for achieving our desired functionality with high precision, while useful information has been investigated in Makki et al. (2016).

## References

- Aivazoglou M, Roussos O, Ioannidis S, Spiliotopoulos D, Polakis J (2017) Reveal: fine grained recommendations in online social networks. In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2017), Sydney, pp 1–4
- Antonakaki D, Spiliotopoulos D, Samaras CV, Pratikakis P, Ioannidis S, Fragopoulou P (2017) Social media analysis during political turbulence. *PLoS ONE* 12(10):1–23
- Bakshy E, Eckles D, Yan R, Rosenn I (2012) Social influence in social advertising: evidence from field experiments. In: Proceedings of the 13th ACM conference on electronic commerce, Valencia, pp 146–161
- Berkovsky S, Freyne J (2015) Web personalization and recommender systems. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, pp 2307–2308
- Bird S, Loper E, Klein E (2009) Natural language processing with Python. O'Reilly Media Inc, Sebastopol
- Cai X, Bain M, Krzywicki A, Wobcke W, Sok Y, Compton P, Mahidadia A (2010) Collaborative filtering for people to people recommendation in social networks. In: Advances in artificial intelligence. Lecture notes in computer science, vol 6464, pp 476–485
- Camacho L, Alves-Souza SN (2018) Social network data to alleviate cold-start in recommender system: a systematic review. *Inf Process Manag* 54(4):529–544
- Chamoso P, Rivas A, Rodríguez S, Bajo J (2018) Relationship recommender system in a business and employment-oriented social network. *Inf Sci* 433:204–220
- Chin J, Diehl V, Norman K (1988) Development of an instrument measuring user satisfaction of human-computer interface. In: Proceedings of the '88 SIGCHI conference on human factors in computing systems, Washington, DC, pp 213–218
- Contratres FG, Alves-Souza SN, Filgueiras L, DeSouza L (2018) Sentiment analysis of social network data for cold-start relief in recommender systems. In: Proceedings of the 2018 conference on information systems and technologies, Naples, pp 122–132
- Da' u A, Salim N, Rabi I, Osman A (2020) Weighted aspect-based opinion mining using deep learning for recommender system. *Expert Syst Appl* 140:112871
- De Meo P, Fotia L, Messina F, Rosaci D, Sarné GM (2018) Providing recommendations in social networks by integrating local and global reputation. *Inf Syst* 78:58–67
- De Pessemier T, Dooms S, Roelandts J, Martens L (2011) Analysis of the information value of user connections for video recommendations in a social network. In: Proceedings of the 2011 CEUR workshop, Pisa, pp 1–7
- Dridi A, Reforgiato Recupero D (2019) Leveraging semantics for sentiment polarity detection in social media. *Int J Mach Learn Cybern* 10:2045
- Facebook (2017) Graph API edges. <https://goo.gl/PdWuYm>. Accessed 11 Oct 2018
- Facebook (2018) Using actions. <https://goo.gl/9schoS>. Accessed 15 Oct 2018
- Facebook (2019) Facebook—how does news feed decide which stories to show? <https://goo.gl/Dupvg8>. Accessed 20 Oct 2019
- Facebook news feed (2016) News Feed FYI: a window into news feed. <https://goo.gl/ByPLgF>. Accessed 19 Oct 2018
- Fang X, Zhan J (2015) Sentiment analysis using product review data. *J Big Data* 2(1):1–14
- Freebase (2018) A community-curated database of well-known people, places, and things. <https://developers.google.com/freebase/>. Accessed 11 Nov 2018
- Garosi F (2008) PyCLIPS Manual. <http://pyclips.sourceforge.net/manual/pyclips.html>. Accessed 11 Mar 2019
- Gretarsson B, O'Donovan J, Bostandjiev S, Hall C, Holzer T (2010) Smallworlds: visualizing social recommendations. *Comput Graph Forum* 29(3):833–842
- Guy I, Zwerdling N, Ronen I, Carmel D, Uziel E (2010) Social media recommendation based on people and tags. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval (SIGIR '10), New York, NY, pp 194–201
- He J, Chu WW (2010) A social network-based recommender system (SNRS). *Ann Inf Syst* 12:47–74
- Horn L (1989) A natural history of negation. University of Chicago Press, Chicago
- Jones AM, Arya A, Agarwal P, Gaurav P, Arya T (2017) An ontological sub-matrix factorization based approach for cold-start issue in recommender systems. In: Proceedings of the 2017 international conference on current trends in computer, electrical, electronics and communication, Mysore, India, pp 161–166
- Kalaï A, Zayani CA, Amous I, Abdelghani W, Sèdes F (2018) Social collaborative service recommendation approach based on user's trust and domain-specific expertise. *Fut Gen Comput Syst* 80:355–367
- Li J, Yang Y (2018) Recommender systems based on opinion mining and deep neural networks. In: MATEC web of conferences, 173, Article 03016
- Liu B (2015) Sentiment analysis: mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge
- Ma X, Ma J, Li H, Jiang Q, Gao S (2018) ARMOR: a trust-based privacy-preserving framework for decentralized friend recommendation in online social networks. *Fut Gen Comput Syst* 79:82–94
- Makki R, Soto AJ, Brooks S, Milios E (2016) Twitter message recommendation based on user interest profiles. In: Proceedings of the 2016 IEEE/ACM international conference on advances in social networks analysis and mining, San Francisco, pp 406–410
- Margaris D, Vassilakis C (2018) Exploiting rating abstention intervals for addressing concept drift in social network recommender systems. *Inf Multidiscip Digit Publ Inst* 10(7), Article 230
- Margaris D, Vassilakis C, Georgiadis P (2016) Recommendation information diffusion in social networks considering user influence and semantics. *Soc Netw Anal Min* 6(108):1–22
- Margaris D, Vassilakis C, Georgiadis P (2018) Query personalization using social network information and collaborative filtering techniques. *Fut Gen Comput Syst* 78(1):440–450
- Margaris D, Spiliotopoulos D, Vassilakis C (2019) Social relations versus near neighbours: reliable recommenders in Limited Information Social Network Collaborative Filtering for online advertising. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2019), Vancouver, pp 1–8

- Mohammadi SA, Andalib A (2017) Using the opinion leaders in social networks to improve the cold start challenge in recommender systems. In: Proceedings of the 3rd IEEE international conference on web research, Tehran, pp 62–66
- Nazir F, Ghazanfar MA, Maqsood M, Aadil F, Rho S, Mehmood I (2019) Social media signal detection using tweets volume, hashtag, and sentiment analysis. *Multimedia Tools Appl* 78:3553
- Pasricha H, Solanki S (2019) A New Approach for Book Recommendation Using Opinion Leader Mining. *Emerging Research in Electronics, Computer Science and Technology*, Springer-Singapore, pp 501–515
- Polanyi L, Zaenen A (2006) Contextual valence shifters. *Comput Attitude Affect Text Theory Appl* 20:1–10
- Reshma R, Ambikesh G, Thilagam P (2016) Alleviating data sparsity and cold start in recommender systems using social behaviour. Proceedings of the 2016 international conference on recent trends in information technology, New Jersey, pp 1–8
- Sanders N (2011) Twitter sentiment corpus. [https://github.com/zfz/twitter\\_corpus](https://github.com/zfz/twitter_corpus). Accessed 5 Dec 2019
- Sangeetha J, Prakash V (2019) Improved Feature-Specific Collaborative Filtering Model for the Aspect-Opinion Based Product Recommendation. *Advances in Big Data and Cloud Computing*, Springer-Singapore, pp 275–289
- Sauri R (2008) A factuality profiler for eventualities in text. PhD dissertation, Brandeis University
- Shen RP, Zhang HR, Yu H, Min F (2019) Sentiment based matrix factorization with reliability for recommendation. *Expert Syst Appl* 135:249–258
- Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon based methods for sentiment analysis. *Comput Linguist Arch* 37(2):267–307
- Tewari AS, Jain R, Singh JP, Barman AG (2019) Personalized product recommendation using aspect-based opinion mining of reviews. In: Proceedings of the international ethical hacking conference 2018, Kolkata, India, pp 443–453
- Twitter (2019) Twitter social network homepage. <https://twitter.com/>. Accessed 11 Mar 2019
- Urban dictionary (2019) <http://www.urbandictionary.com/>. Accessed 25 Oct 2019
- Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in facebook. In: Proceedings of the 2nd ACM workshop on online social networks, Barcelona, pp 37–42
- Wang F, Jiang W, Li X, Wang G (2018) Maximizing positive influence spread in online social networks via fluid dynamics. *Fut Gen Comput Syst* 86:1491–1502
- Wilson C, Sala A, Puttaswamy KP, Zhao BY (2012) Beyond social graphs: user interactions in online social networks and their implications. *ACM Trans Web* 6(4):1–31
- Wired (2013) Why Facebook is teaching its machines to think like humans. <https://goo.gl/wk365x>. Accessed 5 Dec 2018
- Zhang K, Cheng Y, Xie Y, Honbo D, Agrawal A, Palsetia D, Lee K, Liao WK, Choudhary A (2015) Ses: sentiment elicitation system for social media data. In: Proceedings of the 11th international conference on data mining, Las Vegas, pp 1–11
- Zhang Q, Wu J, Zhang Q, Zhang P, Long G, Zhang C (2018) Dual influence embedded social recommendation. *World Wide Web* 21(4):849–874
- Zheng X, Luo Y, Sun L, Zhang J, Chen F (2018) A tourism destination recommender system using users' sentiment and temporal dynamics. *J Intell Inf Syst* 51(3):557–578

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.