Reveal: Fine-grained Recommendations in Online Social Networks

Markos Aivazoglou[1], Antonios O. Roussos[1], Sotiris Ioannidis[1], Dimitris Spiliotopoulos[1], and Jason Polakis[2]
[1]FORTH-ICS, email: {markos, roussos, sotiris, dspiliot}@ics.forth.gr
[2]University of Illinois at Chicago, email: polakis@uic.edu

***Abstract*—Content selection in social networks is driven by numerous extraneous factors that can result in the loss of content of interest. In this paper we present `Reveal`, a fine-grained recommender system for social networks, designed to recommend media content posted by the user's friends. The intuition is to leverage the abundance of pre-existing information and identify overlapping user interests in specific sub-categories. While our system is intended as a component of the social network, we develop a proof-of-concept implementation for Facebook, and explore the effectiveness of our underlying mechanisms. Our experimental evaluation demonstrates the effectiveness of our approach, while our user study highlights how `Reveal` offers functionality that could significantly improve user experience.**

I. Introduction

Online social networks are among the most popular web services, with reports indicating that they have also become the most-time consuming user activity [1]. A critical aspect of their popularity, which drives user engagement, is the sharing of content among friends. However, the massive amount of content published [2] can result in users missing content of interest, due to the overall noise. Furthermore, the content shown in Facebook's News Feed is heavily influenced by the user's relationship to the friend that published it [3]; thus, "weak" social ties can lead to the loss of content of interest.

In this paper we propose `Reveal`, a recommender system that exists within the social network that categorizes content and offers suggestions based on the interest similarities between the user and the friend that published the content. Specifically, `Reveal` processes all content published by the user's friends, identifies content from different categories (e.g., music), and then collects information to assign that content to a more specific sub-category (e.g., music (sub)genres). Subsequently, the system analyzes accompanying text to identify the sentiment regarding that content and infer if a positive or negative opinion was expressed. Based on the user's *interest profile*, and the *similarity score* with that friend, `Reveal` determines if the content should be suggested to the user or not.

Previous work has focused on `like` objects to recommend content to users [4]. Similarly, Facebook has also implemented a recommender system that suggests content liked by friends. Our system follows a different approach; it aims to filter through the massive amount of content that is posted by a user's friends, and select posts 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 Latin Jazz music, but disagree when it comes to contemporary classical music.

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, each dedicated to a specific category, thus expanding the existing functionality of the social network, and improving the user experience. The main contributions of this work are the following:

- We design, develop, and evaluate `Reveal`, a novel system designed to recommend the most relevant and interesting media published by a user's social graph.
- We conduct a qualitative user study regarding how Facebook users perceive the value of social recommendations and expectations of their social circle's ability to provide media content recommendations, and identify specific requirements for recommender system design within a network of friends.
- We conduct a three week pilot study with Facebook users to assess our prototype using real-time data, with user feedback validating both the effectiveness of our content selection approach, and the overall usability of `Reveal`.

II. Problem definition, User Study and Motivation

Apart from human interaction in social networks being naturally limited by time constraints, Wilson et al. [5] found that for the vast majority of users, $\sim$70% of their interactions is directed to only 20% of their friends. Viswanath et al. [6] found that only 30% of Facebook user pairs interact consistently at least once every month. These findings support the motivation for `Reveal`, 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. De Pessemier et al. [7] 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 having many friends can further skew the content recommended in Facebook, as content selection is biased towards popular posts.

To obtain further insight regarding the desired characteristics of social recommendations, we conducted a user study with N=38 participants (age group = 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 users perceive their time spent on Facebook, as a recommendation medium for entertainment value (specifically for music and movies). Additionally, user relations were explored to gain insight into the potential benefit of using social data for movie and music recommendations.

Procedure. The participants were recruited through 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.) Moreover, an extensive set of questions constructed for this study was administered to collect interaction-specific insights and the user expectations/requirements for the proposed fine-grained recommendation approach. This included a concrete approach towards functional requirements and social interaction habits. The data was collected via online forms and a follow-up remote debriefing using validation questions to address possible false positives and solidify true positives.

User profiling and general information. To better understand user interaction and expectations for the proposed solution, participants were requested to provide feedback on how 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=0.033). 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=0.022). 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=0.039 and $\chi^2(6)$=15.453, p=0.017, 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=0.007). 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=0.014). 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=0.027). Thus, it is clear 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 (i.e., content *genres*), and not based on the number of overall likes or strength of the social connection.

Interaction. 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 everyday in the network ($\chi^2(12)$=16.158, p=0.007). 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=0.009). 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 ($\chi^2(3)$=9.491, p=0.023) and to edit and fine-tune the recommendations for improved accuracy ($\chi^2(3)$=8.73, p=0.033). The later ability is also very important to participants that check their News Feed infrequently (z=3.089, p=0.002). 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=0.002), illustrating the expected positive impact of the particular functionality within a social network.

Expectations for the proposed solution. Users that do not currently 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, and consider the option to view a collection of recommendations as very useful ($t(36)$=2.782, p=0.009). Interestingly, 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 (assigning 5) the usefulness of viewing a recommendation collection, i.e., filtering out the noise from unrelated posts (z=2.443, p=0.015). 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=0.004), as well as the time spent on the News Feed ($\chi^2(12)$=16.158, p=0.011). Regarding specific expectations, the participants expected overlapping interests with a relative sub-group of friends and stressed the importance of choosing to extract the information from those specific sub-groups to result in accurate recommendations ($F(2,35)$=3.606, p=0.020). The ability to view ranked recommendations that have highly relevant interests was a major requirement, aligning 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=0.006). These findings guided our design:

- Adopt user-driven approach, backed by intuitive social analytics, that leverages overlapping interests with friends.
- Allow users to leverage their existing knowledge of their social circle and easily modify the weights that drive the content selection process, to better reflect interest overlap.

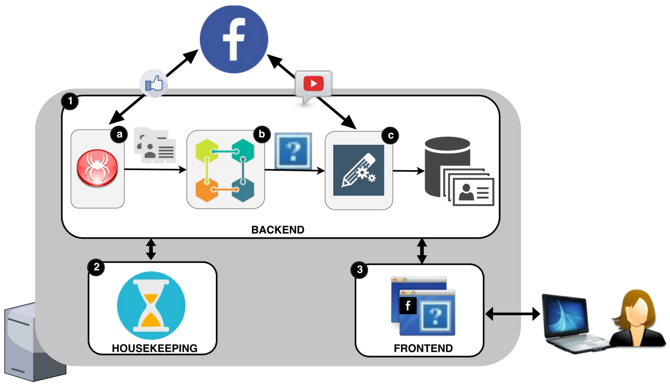

Fig. 1. Overview of main components and workflow of `Reveal`.

III. Methodology

Here we present a high-level overview of our proposed methodology; `Reveal` consists of three main components, as shown in Figure 1.

1 **Backend.** The main component comprises the *backend*, and 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 analyzed, for extracting entities and conducting a fine-grained categorization. This data is then leveraged for creating users' *interest profiles*.

b *Similarity calculation and score tweaking*, in which `Reveal` 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*, for processing the text accompanying content published by the user's friends. This step extracts semantic information from posts and infers whether the poster has expressed a positive or negative opinion about the media content. This allows `Reveal` 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 containing entities of interest, also acting as a data cache to avoid excessive network traffic and operational overhead delays. Our entity extraction mechanism also leverages online resources for data not found locally.

2 **Housekeeping.** This component 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.

TABLE I. Notation used for 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	OveralGenreScore	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	OveralLikeScore	sum of like scores in LS

3 **Frontend.** This is the interactive component of our app, 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.

A. 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*), is collected and processed. We create *interest profiles* from *likes* found on their personal Facebook profiles, namely, under the Music and Movies endpoints [8]. 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[1] and categories. Table I provides an easy reference for our notations used in the bootstrapping phase.

Genre and like scoring. Facebook allows users to express their interests through *likes* associated with specific objects (pages, posts, etc). 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. We create *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. Table II shows an example *GSL*.

TABLE II. Sample Genre Score List for a user with 2 *likes* in her profile.

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

To normalize the genre similarity between the user and her friends at a later stage, we also calculate an overall score of genres for each of them as shown in Equation 1.

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

where $GS(i)$ is the genre score for a specific entry in *GSL*.

[1]For simplicity, *genre* will denote fine-grained sub-genre information.

In addition, we create a *LSL* for every *clique* member's *likes*, to weight them using the *GS* values in *GSL* calculated above. *LSL* is an average calculation of the *GS* corresponding to the genres found both in the *like* item and in *GSL*. For the example of *The Rolling Stones* in Table II, they get a score of 1.25 due to the associated *genres* ($5/4 = 1.25$). This metric allows us to weigh `likes` according to the *genre* preferences of the user. The *LS* scores in *LSL* are used for calculating similarity with other users. We create an overall `Like` score,

$$OLS = \sum_{j \in LSL} LS(j) \tag{2}$$

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*.

B. Sentiment Analysis

Posts are first processed by our sentiment analysis module, and those with negative polarity are discarded. If there is no accompanying text we assume that a positive polarity is implied, the intuition being that users are implicitly recommending them.

Analyser. 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 text, and has been extensively explored in various contexts [9], [10], [11], [12]. `Reveal` needs the ability to discern the poster's opinion, for providing accurate recommendations, and leverages a modified version of the SO-CAL approach introduced by Taboada et al. [13]. SO-CAL extracts 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 for 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 performed better than other dictionary-based approaches we tested. Additionally, we made some modifications, such as changing the SO value of certain words, simplifying the negation lexicon, and adding an emoticon lexicon (110 entries) that we created manually using information from Wikipedia.

Text pre-processing. We filter out Facebook tags (with @) and non-English or non-printable characters. We use the NLTK toolkit for part-of-speech (POS) tagging and the Pattern module [14] for text processing (splitting and tokenization).

Semantic strength tagging using dictionaries. SO-CAL uses dictionaries with words grouped by their Part-of-Speech, ranked with valence strength [15] (in the range $[-5,0) \cup (0,5]$). Tagging is necessary as a word may be defined with different POS, which results in different valence strength [13]. Additionally, as the use of emoticons is widespread in social media, we rank and handle them accordingly.

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 [15]. Their area of effect is limited, which is defined by various grammatical properties. Each valence shifter is assigned an 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 words is 1.

Negation. We apply *shift negation*, where a term's SO value is shifted towards the opposite polarity by a fixed amount.

Scoring. Valence shifters and negators are applied as modifiers to the SO value. To calculate the final SO value, we recursively apply any modifier value found searching backwards, until a determiner (e.g., a *comma* or sentence connective) is found. This calculation is applied to every lexical term and the sum of every sentence's SO value is aggregated, producing the total SO value of a given text.

Irrealis blocking [13], [16], 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 (e.g., modals and quoted sentences), have a nullified SO value. We also ignore sentences that are categorized as questions.

C. Entity Extraction

To gather information about the friends' posts, we have to analyze their entire activity. Our goal is to define all the entities in a friend's post and keep those that belong to categories of interest. `Reveal` utilizes 3 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 [17]. Specifically, we use the *listening* and *watching* social stories.

We also leverage the Freebase [18] 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, that 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 minimal reliance on online APIs, and also reduces excessive network traffic and delays due to operational overhead.

Links. Facebook *Links* are posts that contain a link/URL. To extract entities, we obtain each URL by calling the appropriate Graph API endpoint, filter out non-Youtube URLs, and extract the Video ID. We match potential entities using the aforementioned `Video ID`-to-`Topic ID` mechanism.

Our entities in the knowledge base are indexed with their unique `Topic ID`, enabling us to directly extract any of them that is related to a given Youtube video. Figure 2 presents the workflow of our link analysis.

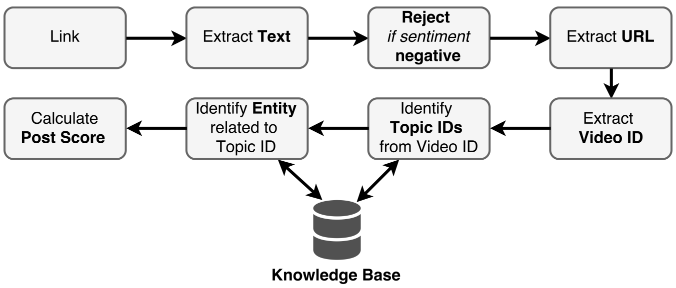

Fig. 2. Workflow of the link analysis process.

Actions. Entity recognition in these posts is straightforward, as the name or title of the item is found within the post's data. Since Facebook prohibits fetching comments accompanying Actions, again we assume they are positive recommendations. While this might not always hold true, our experimental evaluation and user study (Sections III-F, IV) highlight the accuracy of our recommendations; thus, we consider our assumption reasonable. Furthermore, as `Reveal` is designed to be deployed by Facebook, in practice it would also have access to the comments. To find the corresponding entity of an *action* item, we consult our knowledge base using the Facebook object's name. There is no need to identify the category as it is predefined by the Graph API's endpoints.

D. Quantifying Similarity

A challenging aspect of the content recommendation process, is selecting the similarity formula that will quantify how interesting a specific post will be to the user. In previous work [4] the proposed system used an interactive graph, where item and friend weights were assigned manually and required user interaction even in the initial stage. Scores were also in a capped scale (1 to 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 an 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 III. Notation used for similarity calculations.

	Variable	**Description**
GSS	GenreSimilarityScore	Genre similarity between user & friend
LSS	LikeSimilarityScore	Like similarity between user & friend
FSS	FriendSimilarityScore	GSS/LSS similarity between user & friend
PS	PostScore	Rank of a friend's post
PG	PostGenres	Genres extracted from a post

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.

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

The same principle applies for the *LSL* table.

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

Using (3) and (4), we calculate the FSS that the (active) user has with a friend in either category, which represents the actual overlap between them and is used for post score calculation in a later stage. Based on our initial observations, we found that 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 our approach. We find that a weighted average is suitable for treating content classification as a defining factor, and manual testing showed us that giving genre overlap double the weight boosted accuracy in a more fine-grained fashion.

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

Finally, to populate the recommendation lists we gather all the posts contained in the user's friends' profiles and apply the formula shown in (Equation 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.

$$PS = \frac{\sum_{i \in GSL_U(PG)} GS_U(i)}{|GSL_U(PG)|} \cdot FSS_{poster} \quad (6)$$

The resulting item score is not capped, as the values of genre and like scores cannot be predicted, and also allows 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.

E. Facebook Application

We detail our prototype implementation of `Reveal`'s *frontend*, a Facebook Canvas app that leverages the Graph API.

Initialization and score adjustment. Upon installation, `Reveal` first creates the user's *interest profile*. Once it is created, the user is taken to the score adjustment page, with the friend list and similarity scores for each category and genre. Contacts with no overlapping interests are omitted. The user can tweak the scores for each user and genre on a $[0,100]$ scale. This part of our application is presented at the initialization step, and can be accessed at any time for further tweaking the scores.

Top recommendations. 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 two 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*, and recommend three random videos from the current category (one result randomly selected from Youtube, for each of the user's 3 highest ranked genres). 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 matches the user's *interest profile*.

F. Experimental Evaluation and User Study

Named entity recognition. A critical phase of the recommendation process is entity recognition, which allows `Reveal` to identify the entities that the specific post is related to and define the *genres* that are associated with it. To that end, we measure the effectiveness of our module that maps the unique video IDs to unique topic, and thus entity, IDs. We create a dataset by crawling 20 popular music/movie Facebook Group pages. To obtain accurate results we filter out posts that do not contain a valid Youtube URL; our final dataset contains 5,310 links to Youtube. Our module which is able to identify 4,743 valid title-entities, achieving a coverage of 89.32%, demonstrating the robustness of our approach.

Sentiment analysis threshold. Next, we 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 labelled datasets as *ground truth* for evaluating our approach:

- Twitter Dataset [19] (5,513 tweets)
- Facebook Comments Dataset [20] (1,000 comments)

First, we pre-processed the datasets to remove text with a truly neutral sentiment, i.e., 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. In our experiments, we tested 31 different values in the [-1.5,1.5] range as potential thresholds. Nine of those presented significant results, as any value below -1 or above 1 significantly expanded the gap between positive and negative accuracy. As shown

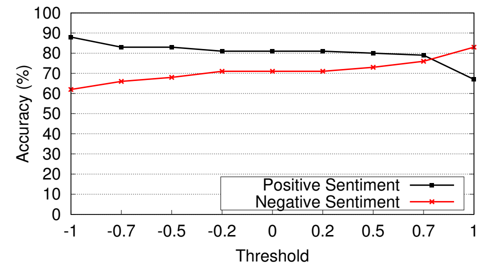

Fig. 3. Sentiment analysis accuracy for different score thresholds.

in Figure 3, values in the range [-0.2, 02] did not present substantial differences, while `Reveal` achieved the best results for a threshold of 0.7, by correctly labeling 79% and 76% of the positive and negative samples respectively. While leverage deep learning neural networks achieve better results (e.g., Socher et al. [21] report up to 85.4% accuracy), our approach is accurate enough to demonstrate the benefits of our fine-grained recommendations; this is also supported by the feedback from the participants in our usability study presented in section IV.

Relevance of content. Next, we wanted to explore how often users' friends publish relevant content that should be processed, and potentially recommended, by `Reveal`. In our initial experiments we created a dataset with 50,000 status updates, but found that our system would extract a very large number of entities that were not relevant to our two categories of interest. This high amount of false positives, and computation overhead, led us to the decision to ignore such posts, and process all posts the belong to one of the following three Graph API Edges: `Links` (any post with an embedded Youtube video), `Watches`, and `Listens`. Table IV breaks down the statistics for relevant content identified by our system's heuristics, extracted by Facebook posts with any of the aforementioned *Edges*, from 3,493 accounts that were connected to our study's 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, 59.98% (94,626 of the 157,762) of the link posts were broken and, thus, removed. Also posts under irrelevant post categories (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.

Reveal vs Facebook. To explore whether users missed relevant content due to Facebook's News Feed personalization algorithm, we gathered every News Feed post from N=7 participants (age M=27.6, SD=3.36; 71.4% male), with number of friends M=547.4, SD=154, that gave us access to their data for a time period of 2 weeks (maximum allowed by Graph API). We then compared that to the content posted by all their friends, and also processed the data with `Reveal` to identify posts of

TABLE IV. 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	1,262
- Movies	28,994
- Music	46,700

TABLE V. Number of recommendations selected by Reveal, compared to the content presented in the users' News Feeds by Facebook.

User	News Feed	Common Items	Additional by Reveal
1	2,335	319	165
2	983	3	481
3	919	120	333
4	968	115	344
5	1,034	117	402
6	12,137	572	531
7	6,170	434	871
Total	**24,546**	**1,680**	**3,127**

interest. We then compared the two datasets and identified the common items. As shown in Table V, while our system selected posts already presented in the News Feed, it also revealed more than twice as many additional posts that are suitable for recommendation as they match the users' *interest profiles*. We cannot assert that all these recommendations would be of interest to the users, or that there was no content of interest overlooked; however, feedback from our usability study presented in the next section offers an initial validation of `Reveal`'s accuracy.

IV. User Evaluation

We conducted a pilot deployment of our prototype as a preliminary usability study, with *N*=5 participants (age *M*=25.4, *SD*=0.89; 80% male), with a large Facebook number of friends (*M*=396.4, *SD*=143), not from the initial user study group (to avoid positive bias, due to our design being guided by that study's findings). They were recruited anonymously, though institutional mailing lists with two requirements: age group 22-34 (for compliance with the initial user study) and >200 friends to allow enough relevant posts for recommendation. The number of participants was low since users were required to evaluate our approach over a long period of time, which would allow them to more accurately assess and evaluate `Reveal`.

The goal was to allow the participants to extensively interact with `Reveal` within Facebook, using their own accounts in a 3-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 QUIS [22] 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 three consecutive weeks. They were asked to

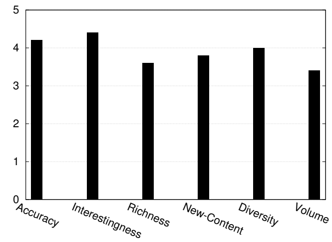

Fig. 4. User feedback on *recommendations* (content) by `Reveal`.

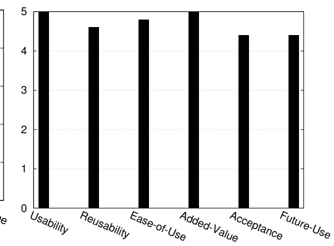

Fig. 5. User feedback on *usability* of `Reveal`.

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 think-aloud protocol in written form. Usability feedback was collected via online forms and the full interaction feedback from assigned online diaries.

A crucial finding was that users were given an adequate number of recommendations, taking into account the number of friends they had. From those, on average, 32% were new posts with more than 50% classified as interesting. As seen in figure 4, users reported very high accuracy and interestingness of video and music posts, 82% and 84% on average, respectively. The new content found in `Reveal` was deemed very adequate and quite rich and diverse. Those qualities were attributed based on the use and revisiting of the application in the three 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.

Feedback on `Reveal`'s usability was positive; as shown in Figure 5, participants easily navigated and used the app for the entire duration of the evaluation. After 3 weeks of use, users reported that they were pretty familiar with the way the app works and easily integrated it 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 recommendations in the future, recognising significant added value from our proposed approach.

V. Related Work

Social Content filtering. Previous work has leveraged social networks for overcoming problems that affect recommender systems, such as cold-start, item sparsity and explicit user ratings [23], [24]. Bellogin et al. [25] presented a Facebook app that investigated user activities that influence the initial item rating and reduce the cold-start problem, while Guangyuan et al. [26] import user information from external social media i.e.,Google+. To enhance users' interests, Shapira et al. [27], enriched their item rating method by exploring the number of user clicks on posted items. Carmagnola et al. [28] used information of related users in social networks, and explored how trending topics among their circle affect their likings. Furthermore, He et al. [29], introduced a more fine-grained rating system using collaborative filtering, and incorporated semantic filtering of social networks. Bonhard et al. [30] proposed the use of ratings for producing suggestions in controlled networks of familiarity. Makki et al. [31] built user interest profiles, for providing suggestions in Twitter.

Trust-Based Profiling. Another issue explored extensively, is how to determine trust between users and leverage that information for improving recommendations. The semi-automatic approach proposed by Cakiroglu et al. [32] used social circles along with selected trusted reviewers, while Gretarssonet al. [4] introduced a graph interface for users to apply weights on both preference in items and similarity in taste with their friends. By leveraging user activity and publicly accessible information, trust-based profiles are created automatically, which can boost the recommendation process [33], [34]. Quijano-Sanchez et al. [35] developed a Facebook app that recommends movies in specific Facebook groups, by using them as trust groups and combining interest profiles with the trust information. Furthermore, Jamali [23], proposed trust propagation between directly or indirectly connected users in social networks.

Sentiment analysis. Integrating opinion-mining in recommender systems could provide more accurate information about users' moods and improve topic rating. Pak et al. [36] mined opinions from a plethora of topics using tweets, while Cakiroglu [32] opted for a simple and targeted approach of integrating opinion-mining in comments that users made after they were asked to. Leung et al. [37] incorporated the use of text reviews in collaborative filtering algorithms for movie recommendations. Feltoni et al. [38], investigated the sentiment of user tweets for recommending new user connections.

VI. Conclusion

We explored the utility of a recommender system within a social network, designed to select content published by the user's friends that matches a fine-grained interest profile that is automatically generated from social data. Our approach is designed as a complimentary mechanism to the main content selection algorithm (e.g., News Feed in Facebook), which is driven by the amount of interaction with each friend. While this is an intuitive and effective selection criteria for general content (e.g., life events), it is not optimal for "entertainment" content. Our prototype Facebook app allowed us to explore the practical aspects and intricacies of processing and extracting information from user-published content. Our subsequent user study, asserted the effectiveness of our approach, and the suitability of social networks as a information-rich ecosystem for providing fine-grained recommendations to users.

References

[1] (2013) Emarketer - social, digital video drive further growth in time spent online. https://goo.gl/28sfk7.

[2] (2016) Facebook - news feed fyi. https://goo.gl/ByPLgF.

[3] (2016) Facebook - how does news feed decide which stories to show? https://goo.gl/Dupvg8.

[4] B. Gretarsson, J. O'Donovan, S. Bostandjiev, C. Hall, and T. Holerer, "Smallworlds: Visualizing social recommendations." *Comput. Graph. Forum*, vol. 29, no. 3, 2010.

[5] C. Wilson, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "Beyond social graphs: User interactions in online social networks and their implications," *ACM (TWEB)*, vol. 6, 2012.

[6] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *WOSN*, 2009.

[7] T. De Pessemier, S. Dooms, J. Roelandts, and L. Martens, "Analysis of the information value of user connections for video recommendations in a social network," in *FutureTV-2011*, 2011.

[8] (2016) Facebook - graph api edges. https://goo.gl/PdWuYm.

[9] S. Mohammad, "From once upon a time to happily ever after: Tracking emotions in novels and fairy tales," in *5th ACL-HLT*, 2011.

[10] F. Å. Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs," *CoRR*, vol. abs/1103.2903, 2011.

[11] S. Gouws, D. Metzler, C. Cai, and E. Hovy, "Contextual bearing on linguistic variation in social media," in *LSM workshop*, ser. LSM, 2011.

[12] B. Liu, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the web," in *14th WWW*, ser. WWW, 2005.

[13] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguist.*, vol. 37, 2011.

[14] T. De Smedt and W. Daelemans, "Pattern for python," *J. Mach. Learn. Res.*, vol. 13, no. 1, 2012.

[15] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing attitude and affect in text: Theory and applications*, 2006.

[16] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[17] (2016) Facebook - using actions. https://goo.gl/9schoS.

[18] (2016) Freebase - a community-curated database of well-known people, places, and things. https://developers.google.com/freebase/.

[19] N. Sanders. (2011) Twitter sentiment corpus. https://goo.gl/QwEP68.

[20] K. Zhang, Y. Cheng, Y. Xie, D. Honbo, A. Agrawal, D. Palsetia, K. Lee, W.-k. Liao, and A. Choudhary, "Ses: Sentiment elicitation system for social media data," in *IEEE*, ser. ICDMW, 2011.

[21] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013.

[22] J. Chin, V. Diehl, and K. Norman, "Development of an instrument measuring user satisfaction of human-computer interface," in *CHI '88*.

[23] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *RecSys '10*.

[24] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *SIGIR*. ACM, 2009.

[25] A. Bellogín, I. Cantador, P. Castells, and F. Díez, "Exploiting social networks in recommendation: a multi-domain comparison." in *DIR '13*.

[26] G. Piao and J. Breslin, "Analyzing aggregated semantics-enabled user modeling on google+ and twitter for personalized link recommendations," in *UMAP '16*.

[27] B. Shapira, L. Rokach, and S. Freilikhman, "Facebook single and cross domain data for recommendation systems," *User Modeling and User-Adapted Interaction*, vol. 23, no. 2-3, 2013.

[28] F. Carmagnola, F. Vernero, and P. Grillo, "Sonars: A social networks-based algorithm for social recommender systems," in *User Modeling, Adaptation, and Personalization*. Springer, 2009.

[29] J. He, "A social network-based recommender system," Ph.D. dissertation, UCLA, 2010.

[30] P. Bonhard and M. A. Sasse, "'knowing me, knowing you' – using profiles and social networking to improve recommender systems," *BT Technology Journal*, vol. 24, no. 3, Jul. 2006.

[31] R. Makki, A. J. Soto, S. Brooks, and E. Milios, "Twitter message recommendation based on user interest profiles," in *ASONAM '16*.

[32] S. Cakiroglu and A. Birturk, "Suggest me a movie: A multi-client movie recommendation application on facebook." in *ISCIS*, vol. 62, 2010.

[33] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Web search and data mining*. ACM, 2011.

[34] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, 2012.

[35] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz, "Happy movie: A group recommender application in facebook," in *24th FLAIRS*, 2011.

[36] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREC*, 2010.

[37] C. W. Leung, S. C. Chan, and F.-l. Chung, "Integrating collaborative filtering and sentiment analysis: A rating inference approach," in *ECAI workshop on recommender systems*, 2006.

[38] D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti, "A sentiment-based approach to twitter user recommendation," in *RecSys '13*.