Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 9, No. 1, 2021, 45-57.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research Paper

GroupRank: Ranking Online Social Groups Based on User Membership Records

Ali Hashemi and Mohammad Ali Zare Chahooki*

Software Engineering Department, Yazd University, Daneshgah Street, Yazd, Yazd, Iran.

Article Info

Article History:
Received 04 May 2019
Revised 31 March 2020
Accepted 27 October 2020

DOI:10.22044/jadm.2020.8337.1973

Keywords:
Social Networks, Instant Messenger, Search Engine, Ranking, Telegram.

**Corresponding author: chahooki@yazd.ac.ir (M. A. Z. Chahooki).*

Abstract

Social networks are valuable sources for the marketers, who can publish campaigns to reach target audiences according to their interest. Although Telegram was primarily designed as an instant messenger, it is now used as a social network in Iran due to the censorship of Facebook, Twitter, etc. Telegram neither provides a marketing platform nor the possibility to search among groups. It is difficult for the marketers to find target audience groups in Telegram, and hence, we have developed a system to fill the gap. The marketers use our system to find target audience groups by keyword search. Our system has to search and rank groups as relevant as possible to the search query. This paper proposes a method called GroupRank to improve the ranking of group searching. GroupRank elicits associative connections among groups based on membership records they have in common. After a detailed analysis, five group quality factors are introduced and used in the ranking. Our proposed method combines the TF-IDF scoring with group quality scores and associative connections among groups. The experimental results show improvement in many different queries.

1. Introduction

Social networks play a vital role in marketing. A huge amount of valuable information is published on social networks every day. The marketers can use this information to select appropriate target markets. Due to censorship of the social networks like Facebook and Twitter in Iran, the users have been inclined to use Telegram instead. Although Telegram is an instant messenger, its features like super-groups, channels, and bots have made it an adequate alternative for the social networks. A high proportion of social networks revenue is from advertisement. Most famous social networks have developed a built-in advertisement platform to select target audiences and promote posts and pages. To the contrary, Telegram does not have such a platform. As a result, the Iranian advertisers have to negotiate with each group or channel owner individually to publish their campaigns. The advertising costs are arbitrarily set by the owners of groups and channels. Furthermore, the advertisers are uncertain about the efficiency of their campaigns because it is easy to cheat on statistics using bots and fake users. In order to mitigate these problems, we developed a system called IdeKav, which analyzes Telegram for the advertisers. Our goal is to fill the vacant place of targeted advertisement feature in the Telegram environment.

In this paper, we propose a scalable on-the-fly method to improve the ranking of social groups called GroupRank. Predominantly, GroupRank takes group membership records into consideration. Similar to the basket analysis algorithms, GroupRank tries to find associations among the users. Naturally, a user wishes to join a group whose activity is in line with his/her activity. Therefore, we used group membership records in this research work to find similar users in a group and rank the group. If we already know that group X is a top result, similar groups to X deserve a boost in ranking.

The remainder of this paper is organized as what follows. The research objectives are demonstrated in Section 2. In Section 3, we review the related works. Section 4 is dedicated to demonstrate the GroupRank algorithm. In Section 5, the empirical results are discussed. Section 6 is the conclusion of our work along with a view for the future.

2. Research Objectives

Telegram is currently the most popular instant messenger in Iran. Many users prefer to share and view content in Telegram rather than anywhere else on the web. The Telegram channels and bots have facilitated public content sharing. The Telegram super-groups can have up to 200000 members. A huge amount of valuable content is shared in super-groups every day. Therefore, the marketers are keenly interested in advertising in Telegram. However, they are not able to analyze the network and find their target audience group easily. Currently, Telegram provides no marketing solution.

The general objective of this research work is a method to find the best possible target audience group for the marketers based on the Telegram groups' information. The better we select the target audiences for a campaign, the better will be the conversion rate. Our system focuses on individual interests according to their behavior on Telegram. The popularity of Telegram among the Iranian society has led to the creation of thousands of groups on diverse topics. One of the most salient features of our system is group search since Telegram has no search option for groups. The channel searching capability in Telegram is very limited and restrictive. We used Lucene TF-IDF (Term Frequency–Inverse Document Frequency) to rank groups in our initial release. In the initial release, the top 5 results were usually good but a lot of relevant groups were not retrieved even in top 100. The technical objective of this research work is to improve the Telegram groups ranking in search.

3. Related Works

In contrast to mass marketing, targeted marketing emphasizes on the precision of the intended audience, which leads to a higher return-on-investment (ROI), especially with the aid of data abundance and rapid progress of data science in the recent years. In targeted marketing, we intend to spend a limited budget on the audiences who are as relevant as possible to the campaign objective. The user profile and interactions are major sources to mine and identify marketing audiences. Mining users' interests from online social networks has been an emerging research topic in the recent years. The knowledge can then be used in friendship prediction, product recommendation, and other marketing purposes. The users' information may also be used in personality and behavior analysis (see [23]). The personality theory claims that a user's personality substantially influences preference. A personality-based product recommender has been proposed in 2017 [24]. The social media data is analyzed in order to predict a user's personality, and to subsequently derive the personality-based product preferences. Chonghuan Xu [11] has proposed a recommendation method based on the social networks. The users' preferences, social relationships, and associations between the users and items are all considered in the similarity computation. The matrix factorization technique is used to alleviate data sparsity and cold-start problems. A hybrid clustering algorithm that composes of K-harmonic means (KHM) and Particle Swarm Optimization (PSO) is then used to obtain a more accurate classification. The benefit of this clustering is that it overcomes the sensitivity of the initial conditions. In a people-to-people recommendation paper [15], a coupled matrix factorization model has been described. The model is used to generate people to people recommendation by utilizing the users' interaction

with items. The Sajad Ahamadian's proposed method [12] is another recommender system based on the users' relations graph. In order to create the initial graph, the nearest neighbors of users are found by means of a clustering algorithm. An iterative process is then applied to the graph to form better cluster centers. An adaptive neighbor selection mechanism along with a confidence model is proposed to prune low-quality neighbors. In a similar work (see [29]), the significance of each user is calculated considering the neighbors. The trusted neighbors of a user are identified and aggregated. Hence, a new rating profile can be established to represent the preferences of the user. By taking an overall look at these research works, we can conclude that the user preference is an important factor in the recommendation. The social networks information is analyzed and processed in different ways in order to obtain the user preference as accurately as possible.

Many researchers believe that people who share similar interests might have different feelings or opinions about them. Recommendation based on the sentiment analysis is rooted in such a fact. In 2018 [13], a recommendation engine relying on identification of semantic attitudes was proposed. The sentiment, volume, and objectivity extracted from the user-generated content are used to create a 3D matrix. Matrix factorization is applied to make recommendation possible at a large scale. Semantic concept clustering has been proposed in the Hong Zhang's work [14]. WordNet and HowNet along with a domain professional dictionary and DMOZ are used to construct semantic relations and a hierarchical system of classification. The user interest model is built on ontological concepts to improve the diversity of recommendation. Siaw Ling Lo *et al*. [22] have developed a combination of semi-supervised and supervised learning methods in order to find high-value social audiences. The term "high-value social audiences" is defined as a segment of online audiences who are interested in the current business plan that is different from influencers in the domain since the latter consists of authoritative people who may or may not be a follower of the account owner. Fuzzy match and Twitter latent Dirichlet allocation are used to group different words with the same meaning in tweets. A vector space is then created for each tweet, and an ensemble of SVMs learns them. Tweets posted by the followers of the account owner are then scored by SVMs, and top audiences get ranked with one of 3 different methods selectively. This approach finds high-value social audiences only among the followers of the account owner. Since each tweet of each audience has to be scored by an ensemble of SVMs (which is time- and resource-consuming), this approach works on the limited followers of the account owner but cannot be scaled up to cover almost all Twitter accounts.

Fattane Zarrinkalam [16] has argued that most existing approaches heavily rely on explicit user-generated content and overlook implicit interests. In order to infer the users' implicit interests, a graph-based link prediction schema that operates over a representation model has been proposed. The representation model consists of the user explicit contributions to topics, relationships between users, and the relatedness between topics. Finally, implicit interests are inferred based on the homophily principle and heterogeneous nature of the graph. The same argument has been made by Vahideh Nobahari [28]. The users' trust, sequential interest, and implicit interest are all considered in their proposed method that is based on matrix factorization. Majed Alrubaian's work [21] measures the users' reputations and analyzes sentiments in order to identify the credible users. The approach consists of two parts, i.e. sentiment analysis and popularity measurement. The sentiment analysis is based on a pre-defined list of negative and positive words. The popularity measurement is based on the parameters like the number of user's followers, retweets, and mentions. Finally, the two scores are combined, and the users are ranked based on the final score. The experimental results of this research work showed that 96% of credible users had no mentions in their tweets, whereas 46% of non-credible users had at least two mentions. Non-credible tweets tend to have at least one hashtag. The credible users also embed some mentions and hashtags in their tweets but their mentions and hashtags exhibit a more stable distribution with respect to the topic or event than the non-credible

users. Another interesting result is that the non-credible users send more tweets, and are represented on more lists than the credible users. Moreover, the non-credible users are more likely to be negative in sentiment, whereas the credible users are more positive.

The recommender systems, in general, suffer from the cold-start problem more or less. Several studies to mitigate the cold-start problem have been reviewed in a paper published in 2018 [17]. The final results showed that few research papers currently use knowledge from the social networks to mitigate the problem. The recommender systems are a sub-field of information retrieval. Many other sub-fields of information retrieval such as search engines do not suffer from the cold-start problem. In the search engines, the user requests are initiated with an explicit query. The retrieved results are ranked based on several factors, and have returned to the user subsequently. Ranking in search engines is of paramount importance, and a lot of efforts have been made in the field of ranking so far. The content-based algorithms such as TF-IDF (see [7]) and BM25 (see [6]) are seminal works in the field, which give weights to words in the content and search user query on an inverted index. The content-based algorithms are good choices to search a library of books but they are not good enough to search web pages or social media content today. The World Wide Web contains a huge amount of spam content. Moreover, the scale of the web is incomparable to the scale of books in a library. The problems with content-based algorithms have led to the development of complementary methods. Many of these methods focus on the links between entities. PageRank is a well-known example of ranking based on links. It propagates a web page score to linked pages in a graph. TwitterRank (see [8]) is another link-based method to find the influential users on Twitter. It propagates influence in the user graph by a random walker, the same as PageRank. The random walker in TwitterRank is topic-sensitive though. TURank (see [9]) considers the number of tweets and retweets along with follows in a heterogeneous graph called Tweet-User graph. Propagations are done in the Tweet-User graph to calculate the final score. These algorithms assume that the links are permanently pointing to the same page or user, which is almost true for web pages and social media. Administrators of the Telegram groups tend to change the group link periodically, though, based on our observations. Based on this fact, such algorithms are not useful in the Telegram environment.

Getting the user feedback is another complementary method widely used to improve the initial content-based ranking. The researchers at Microsoft Research (see [1]) have developed a method incorporating implicit feedback to improve the accuracy of a competitive web search. They used a supervised machine learning technique to learn a ranking function that predicts relevance judgments. They monitored the real users' activities and included click-through features, browsing features, and query-text features in their method. After a detailed analysis, Thorsten Joachims *et al*. [5] claimed that the users' clicking decisions are influenced by the relevance of the results but they are biased by the trust users have in the retrieval function, and by the overall quality of the result set. By the way, it is not possible to collect many such user interactions in Telegram as a third party.

The Google researchers (see [2]) have tried to learn from the user interactions in the personal search by attribute parameterization. They projected the user queries and documents into a multi-dimensional space of fine-grained and semantically coherent attributes. By creating an attribute-attribute graph derived from the document-attribute graph and query-attribute graph, they were able to parameterize the attributes. Private user files are not shared across the users so basically, it is not possible to get help from collaborative knowledge. Attribute parameterization enables an effective usage of cross-user interactions to improve personal search quality.

As a third party in a social network, it is usually possible to crawl pages along with the followers. This can be useful in order to elicit the user's behavior. Fredrik Erlandsson *et al*. (see [3]) have extracted the association rules from Facebook to find the influential users. They stated that the rule-based ranking of the users has a lower execution time compared to the state-of-the-art methods. To

the contrary, they removed the Facebook pages such as Fox News with 837,176 users, 4485 posts, 6,967,304 comments, and a lifetime of 2034 days (almost six years) because they could not calculate the association rules for them using a server with 144 GB of RAM and a 24-core CPU. The association rule extraction is memory- and time-consuming, in general. The algorithms like Dist-Eclat and BigFIM (see [4]) try to speed up and scale out the rule extraction but with a large number of transactions; it is still not possible today to extract the rules on the fly. However, ranking in search is expected to be done in a fraction of a second. In spite of scaling limitation, several recent methods make use of frequent pattern of the mining and association rules. Another researcher [26] has proposed a 3-step approach to detect the users' interest based on the frequent pattern mining. First, the users' explicit interests are inferred by extracting information from the content they have shared. The frequent patterns are then generated based on the collective set of the users' explicit interests (represented as sets of tags). The frequent patterns show the relationship between topics. Finally, additional implicit interests are combined with the frequent patterns learned. The underlying algorithm behind Seyed Ahmad Moosavi's work [25] is very similar to our method. Harmonious groups of the users are obtained by frequent pattern mining of the users' actions. It leads to the extraction of a large number of groups. Most of these groups are either very small or separate. A separate group is a group whose users have a low degree of connection in the social graph. After pruning small and separate groups, the remaining groups are expanded. It is assumed that the neighbors of a group will follow it. Group expansion works are based on this assumption, and virtually assign neighbors to each group.

Ming Yan *et al*. [10] have suggested a cross-network association-based solution for the YouTube video promotion. They first performed a heterogeneous topic modeling, and then applied cross-network topic association. Their work was based on the idea that if many overlapped users who take interests in a YouTube topic also follow a Twitter topic, the association between the two topics tends to be strong. Jiangning He *et al*. [27] have studied the cross-network relationship identification as well. They extracted a series of discriminant features from each social network and merged multiple social networks in terms of features and social links. Then they set an initial influence using a classifier. The initial influence was propagated through a random walk model utilizing the structural information. Finally, a merged social network was obtained. Although they did not explicitly report, their results showed that a unique user may have totally different friends in different social networks. Therefore, the cross-network analysis can be very helpful in identifying the user's preferences.

Preethi Lahoti [18] has proposed a method to find topical experts in Twitter via query-dependent personalized PageRank. Given a text query, the algorithm uses a dynamic topic-sensitive weighting scheme, which sets the weights on the edges of the graph. Then it uses an improved version of query-dependent PageRank in order to find important nodes in the graph, which correspond to the topical experts. Evaluation of a number of different topics demonstrated that the method was competitive with the Twitter's own search system while using less than 0.05% of all Twitter accounts. Richang Hong's work [20] is another graph-based algorithm used to rank the users within a specific time period. The ranking is based on the user vitality. Based on their definition of user vitality, if a user has many interactions with his friends within a time period and most of his friends do not have many interactions with their friends simultaneously, it is very likely that this user has a high vitality. An undirected graph with the users as vertices and interactions as edges is created. The number of interactions between two users in the selected time period is the weight of the edge between the two corresponding vertices in the graph. This type of edge weighting represents a community, where each one of both users makes the same contribution to the interactions that may not be true in reality. For instance, one of the users may be very active to interact, while the other one is relatively passive. Therefore, instead of equally allocating the interactions between two users, it might be better to allocate them according to their vitality. An iterative algorithm runs over the initial

graph to update the edge weight allocations until a stop criterion is satisfied. Although the convergence of iterations is uncertain, the edge weight allocation will approach stability as many iterations happen. By means of this iterative algorithm, the vitality score of a user is not determined only by its own first-level neighbors anymore. The scores of the users in the whole network will influence every single user.

A supervised seeded PageRank algorithm has been proposed recently in order to identify and target the marketing audiences more precisely [19]. The solution is based on utilizing the anonymized interactions of the users by which the users are scored according to their relevance to the marketing campaign objective. The links between two users are weighted with the weights learned in a supervised setting in order to ensure a high relevance to the score prediction task. A seeded variant of PageRank has been modified to adapt to this solution while maintaining the convergence property. Previously, successful marketing targets were treated as seed users to infer a set of good candidates for marketing who may have similar qualities to those seeds. An additional advantage of the inbound-normalized seeded PageRank is that the output scores can be appropriately interpreted as the classification probabilities. This property makes the method easy to combine with the traditional supervised learning approaches.

4. GroupRank

Our proposed method, GroupRank, is a complementary ranking method with TF-IDF as the baseline. It extracts the associative connections among the users of online social groups to improve ranking in search.

We designed a Telegram group crawler in order to test our proposed ranking method with enough data. Telethon[1] library was used for the MTProto[2] connections. Each instance of our crawler is just like a normal Telegram account. The instance follows the links it finds to join new groups. Each group can have up to 100000 members. Due to the Telegram limitation of 500 groups per account, we had to use more than 200 accounts for data collection. We covered about 150,000 Persian-speaking groups sharing about 20 million messages daily.

Table 1 shows the statistics we extracted from Telegram during 3 months of data collection.

Table 1. Telegram statistics extracted from IdeKav.

Number of non-spam messages received	300 million
Average number of non-spam messages daily shared per group	23
Average message length across all groups	174 characters
Number of users covered	37,999,428
Number of groups covered	142,288
Average members of groups	659
Average number of groups each user is a member of	2.46
Average number of groups each user is a member of (if already is a member of at least one group)	3.64
Number of users who left all groups we know	12,228,756
Number of users who left at least one group	25,900,803
Number of groups a user leaves on average in 3 months	2.2
Number of groups a user leaves on average in 3 months (if he/she has left at least 1 group)	3.23
If a user is an administrator, on average, how many groups he/she owns	1.25

The members of a Telegram group can add their contacts to the group. The new members can report this activity as spam if they feel that they are added to an unwanted group. In order to prevent being marked as spam, the current members of the group usually choose people they add carefully. As a result, this feature helps groups grow with more interested audiences.

Telegram does not provide any search functionality for groups. To the best of our knowledge, there are no third-party search engines for the Telegram groups yet either. As the users cannot search for new groups to join globally, they remain in a local cluster of groups. The users of a local cluster are expected to have similar interests because of the Telegram strict membership spam policy. Our proposed algorithm utilizes this connection between the users implied by their membership in groups.

[1] https://github.com/LonamiWebs/Telethon
[2] https://core.telegram.org/mtproto

Based on our observations, we believe that there exist some association rules among the users of a local cluster. These rules can be used to improve the ranking of groups in the search. The idea behind the proposed algorithm is similar to the basket analysis ideas.

2- The query is searched on the data index.
3- Top results are sent to the association index.
4- Top association candidates are returned with the corresponding score.
5- All results are re-ranked based on the quality and return to the end user.

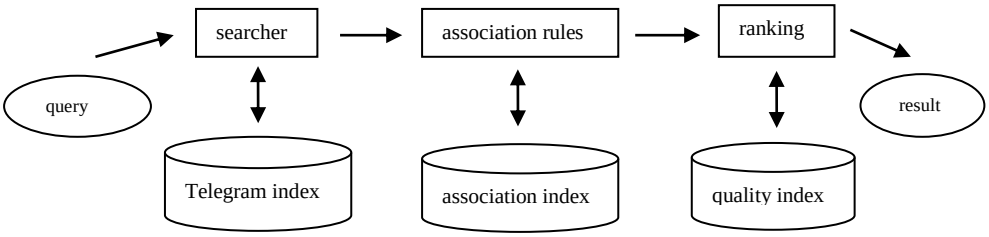

Figure 2. Overall architecture of GroupRank.

Assume that we know the rules like "if a user is a member of groups A, B, and C he/she is interested to join group D." Note that these rules are query-independent and are extracted only based on memberships. In the scope of searching, such rules can be integrated into the system as a factor of ranking. If groups A, B, and C are the top results of a search, group D deserves to get a score boost since it is most probably related to the search query based on the community behavior. We call it the association boost in the rest of this article.

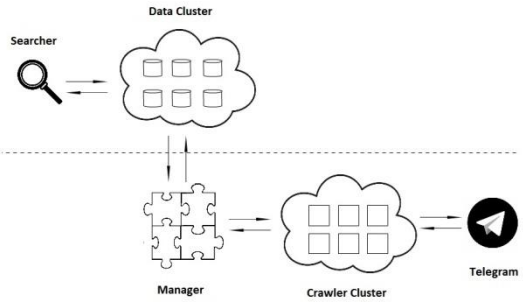

Figure 1. IdeKav core architecture.

Figure 1 is the core architecture of IdeKav. Our crawlers collect and update the membership data of each group regularly. We store them in an indexed structure for a fast retrieval. In order to apply the association boost in ranking, we developed the following procedure. All the steps of the procedure can be finished in less than a second.

1- The end user enters a text query for the search.

Our method is capable of returning the results in a fraction of a second, while the basket analysis algorithms are very time- and memory-consuming. We tried several existing basket analysis algorithms without success. Our dataset was too large for these algorithms to extract the rules in a reasonable time.

Figure 2 is an overview of our proposed method. The data index contains the aggregated text and meta-data of each group. The top results are found based on TF-IDF as the baseline algorithm. Group title, group description, and aggregated messages have score boosts equal to 5, 2, and 1, respectively.

The association index contains the membership data. It is possible to filter a list of groups, users, or both together fast. Table 2 is a sample of the index containing 5 documents. The real index contained more than 177 million documents.

Table 2. A sample of association index.

userID	groupID	status
13850508	15017013	CURRENT_MEMBER
69136668	15017013	FORMER_MEMBER
13850508	37886934	ADMIN
69136668	91534283	CURRENT_MEMBER
37245238	37886934	CURRENT_MEMBER

With a single request, we can obtain a list of memberships similar to a list of purchases in the basket analysis. Each row of the list represents a user and contains the IDs of the groups he/she is a member of. Table 3 shows a sample association list. By considering each user, a purchase, and each group of an item, we can run any basket

analysis or association rule algorithm on this list to extract the rules.

Table 3. A sample association list.

userID	List of groupIDs
13850508	15017013, 37886934
69136668	91534283
37245238	37886934

The association rule algorithms generally require a lot of time and memory. Based on our experiments, extracting the frequent items is enough to calculate the boost scores that make it the least memory- and time-consuming. Note that we do not even need to extract the frequent item sets. Extracting only the frequent items, which is simply counting each group in the list, would lead to an efficient score calculation.

The main reason extracting frequent items are even more accurate than the frequent item sets or association rules in our scenario is that we have filtered our association list by query prior to the analysis. The list only contains the users who are members of at least one of the top groups returned by the query-based search. It makes all the users on the list connected to each other in terms of interest. The other reason is related to the fact that using the items is more natural in the score calculation context than using item sets. In the context of recommendation systems, the item sets help narrowing down the taste of the user. In the score calculation context, we require more generalization rather than specification to calculate the scores, as we have already filtered the list by a query-based search.

In summary, we basically find the associations by running a query that filters a large quantity of association list, and then we just count the remaining groups. The results obtained are similar to the results of running a basket analysis algorithm on the whole list. We demonstrate the procedure by an example.

Suppose that the end user has entered "cryptocurrency" in the search box. First, the query is sent to Lucene. Lucene returns the top groups based on TF-IDF. Table 4 shows the top 5 groups for the aforementioned query.

These results obtained are sent to the association module. The association module uses the association index to retrieve a list like table 2. We call it the association list. Only if a user is at least a member of one of the top 5 groups, he/she is included on the list. The list contains all groups each user is a member of. We count the number of appearances of each group on the list. The more a group is found on the list, the better is the score it gets. Finally, all these groups are re-ranked based on three factors in the final ranking.

Table 4. Top 5 results for "cryptocurrency" query.

groupID	group name	Lucene score
1122822252	Unify Cryptocurrency	9.126912
1395938608	Sandys CryptoCurrency Group	9.009725
1119605452	KRYPTOWOLF-All things cryptocurrency and Online money making	8.620558
1121274918	BITCOIN TRAINING SCHOOL⚡	7.852843
1319242792	Nimecoin.co	7.6586714

The final ranking is the last step of our algorithm. We involve the following 3 factors in our ranking:

1. $S = Query - based\ search\ score$
2. $A = Association\ score$
3. $Q = General\ quality\ score$

In our application, the query-based search score (S) is the TF-IDF score returned by Lucene[3]. We keep this score intact to better evaluate the significance of our algorithm. The association score is calculated based on a couple of factors. The most important factor is the number of times the group is seen in the association list normalized by the total number of group members because generally the more members a group has, the more chance to show up in the association list it gets. We calculate the association score (A) by means of Equation 1. *F* is the frequency of the group in the association list. *U* is the total number of users on the association list.

$$A = \frac{F}{U} \qquad (1)$$

In order to define the group quality factors, we manually looked up and flagged 60 groups as either low or high quality. Most of the groups flagged as low-quality were chit-chat and dating groups. The business and expert groups were flagged as a high quality, in contrast. By

[3] https://lucene.apache.org/

analyzing the behavior and statistics of this small dataset, we defined several factors to assign the quality scores to groups.

In our opinion, the most important factor we observed is Engagement (E). Generally, in the high-quality groups, there were fewer amounts of discussions throughout the day than the low qualities but more people were engaged in each discussion. Each discussion was shorter in the high-quality groups because they were inclined to reach a point or decision as soon as possible. On the other hand, the low-quality groups tend to talk a lot for a long time without a clear objective.

Based on these facts, we defined the engagement factor (E) as the number of active members of the group divided by the number of all messages in the group in a specific period of time. The term active member is referred to any member of the group who has posted at least one message during a specified period of time.

Most high-quality groups have a similar activity pattern. They are highly active in the working hours of the day and almost inactive at midnight. Such a pattern is not seen in low-quality groups as there are some people talking any time of the day.

between 0 and 24, denoting the daily inactivity period for each group. We also noticed that the Closed Hours (CH) is affected by the number of members of the group. The red line in Figure 3 indicates the ratio of groups with CH = 0 in different ranges of members. The ratio increases from 0.06 to 0.39 as the number of members grows. The green area displays the count of groups in each range of members. For example, there are about 45,000 groups with 51 to 100 members, and the Closed Hours (CH) is equal to zero in 9% of them.

Lengthy messages are common in high-quality groups, while messages are usually short in low-quality groups. Among our flagged groups, the average message length was 136 characters for the high-quality groups and 48 characters for the low-quality ones. This fact is evident due to frequent “hi”, “sup”, and similar messages in the low-quality groups, while sharing of long articles is usual in the high-quality groups. The average message length of each group during a specific time period is calculated and called Message Length (ML).

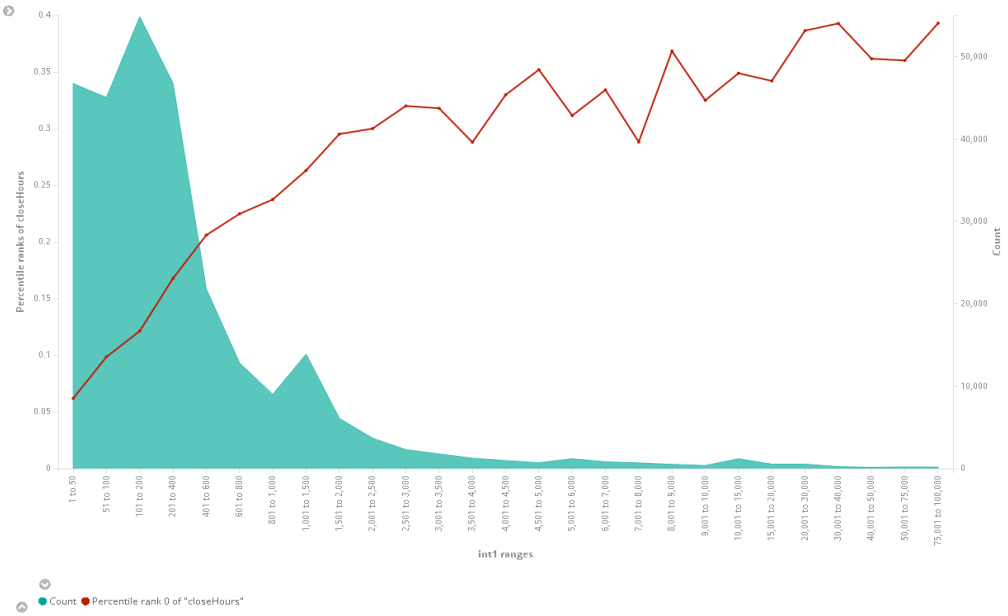

Figure 3. Percentage of groups without closed hours (red) along with the count of the groups in each member range (green).

We measured the daily inactivity periods in hours and named it Closed Hours (CH). It is a number

It is possible to mark and reply to a specific message in a group using the Telegram reply

feature. Replying alerts the sender of the replied message as well as showing a pointer to the new reply. This feature is excessively used in the low-quality groups because there are usually several ongoing discussions at a time. In general, if the Reply Ratio (RR) is small, it means that the messages are unrelated to each other. This behavior is typically seen in free advertisement groups. Most of the members just post their advertisement and leave without paying attention to the other messages of the group. Reply Ratio (RR) in the high-quality groups are neither low nor high compared to all the others.

The low-quality groups tend to use more non-alphanumeric characters both in the group name and the content. Non-alphanumeric Character Ratio (NCR) is another factor we calculate for the group quality. Considering all messages published during a specific time period, the number of non-alphanumeric characters is divided by the number of all characters for each group.

Table 5 briefly demonstrates the aforementioned factors and their impact in the order of their importance. Quality score (Q) is the weighted combination of the impact of all factors. We set the weights subjectively based on our observations and judgments for our experiments. Note that calculation of the general quality score is query-independent and can be done in a periodic iterative offline process.

Table 5. Quality factors and impact.

Factor	Impact	Description
E	Positive	Number of active members of the group divided by the number of all messages in the group
CH	Negative if zero	Inactivity pattern of the group measured in hours
ML	Positive	Average message length
RR	Positive if in IQR	Ratio of replies
NCR	Negative	Ratio of non-alphanumeric characters

Finally, we normalized each score (S, A, and Q) to fit in the $[0,1]$ range. Then we gave a unique coefficient to each one of them. Equation 2 is the final ranking score, where α, β, and γ are the constant coefficients for each normalized score. The coefficients were initially set by intuition and refined with try and error.

$$\alpha S + \beta A + \gamma Q = \text{Final ranking score} \qquad (2)$$

5. Experimental Results

In order to evaluate our proposed method, we examined the results of 100 queries with and without applying our method. We ran these queries in 2 different setups:

1. Lucene TF-IDF search without modification
2. Our proposed method

Table 6 and table 7 show comparison of P@10 and P@50 of the two setups, respectively, for a sample of queries. Figure 4 visualizes P@50 for an easier comparison. Although P@K is a useful metric, it fails to take into account the positions of the relevant results among the top K. We believe that the irrelevant results appearing on top of the search result list should be penalized. Therefore, we devised an evaluation metric to take positions and irrelevant result into consideration too.

Table 6. p@10.

Query	TF-IDF p@10	GroupRank p@10
Cryptocurrency	70	100
Lavender	80	80
Buy hamster food	50	50
Margarita pizza	40	70
Xbox one	70	90

Table 7. p@50.

Query	TF-IDF p@50	GroupRank p@50
Cryptocurrency	46	58
Lavender	66	60
Buy hamster food	24	42
Margarita pizza	36	50
Xbox one	64	72

We stored the results of each setup. Each result had 50 sorted items, and was graded manually by an expert in the field. In order to score a result, the experts had to score each item. Each item had a boost based on its position. Obviously, the items on top had higher boosts than the lower items. Experts had to choose between "relevant", "semi-relevant", and "irrelevant" for each item. Table 8 demonstrates how the score of each result is calculated based on the expert choice. Sum of all boosts of the relevant items was the grade for the result.

Table 8. Expert score calculation (on top-50 results).

Expert choice	Score calculation
Relevant	$(51 - \text{position}) \times 2$
Semi-relevant	0
Irrelevant	$(\text{position} - 51) \times 2$

Table 9 lists a few queries of our experimental results along with the grade experts giving the results of each method. We analyzed each query to get a better understanding of how our method improved ranking. Figure 5 visualizes these results along with the up/down bars.

Table 9. Sample of experimental results.

Query	TF-IDF grade	GroupRank grade
Cryptocurrency	700	1360
Lavender	1482	1236
Buy hamster food	-796	-696
Margarita pizza	-180	178
Xbox one	986	1036

Table 4 shows the top TF-IDF results for the query "cryptocurrency." All the groups on the TF-IDF list have a high frequency of the word "cryptocurrency" in their content. In the top-50 list, there were some groups mainly about investment or job seeking, which were not directly related to cryptocurrency. GroupRank brought up more groups focusing on cryptocurrency, which had a less frequency of the word "cryptocurrency" but had words like "Bitcoin," "Blockchain," and "ICO" in the content. The GroupRank improvement for this query was significant.

In general, the TF-IDF results for "lavender" were good but there was an irrelevant group about bees in the top-5 list. The irrelevant group brought up more irrelevant groups about bees with GroupRank. It led to fewer quality results compared to TF-IDF. This query was the only one (among 100) leading to fewer quality results due to an irrelevant group in the top-5 list. We included the query in the sample in order to show that GroupRank could have an adverse effect on the results if an irrelevant group is in top-5 list.

Due to the paucity of groups focusing on hamsters, the general results for "buy hamster food" were not good with both methods. GroupRank removed a few joking groups from the bottom of the top-50 list causing slightly better results.

The top-5 TF-IDF results for "margarita pizza" were just good. GroupRank used these top candidate results to eliminate the irrelevant results on the top-50 list. There were some spam groups in the TF-IDF top-50 list but they were not in the GroupRank list.

The "Xbox one" results were generally good with TF-IDF but there were some chit-chat groups on the top-50 list. A large proportion of members of the groups focusing on the Xbox one were also members of the chit-chat groups. Due to this fact, GroupRank was unable to remove some of those chit-chats from the top-50 list based on association. Some of those chit-chat groups were removed due to a low quality score. The GroupRank results were slightly better than TF-IDF in this case.

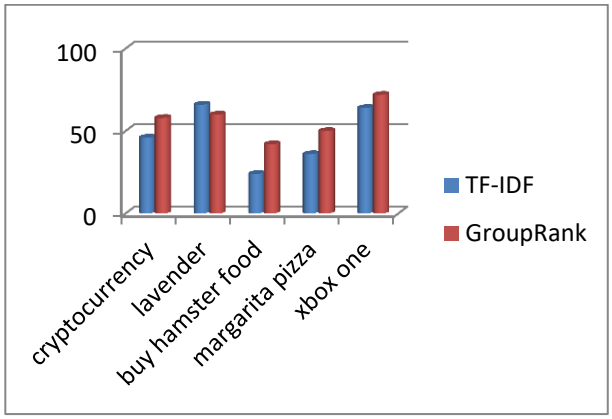

Figure 4. p@50 comparison.

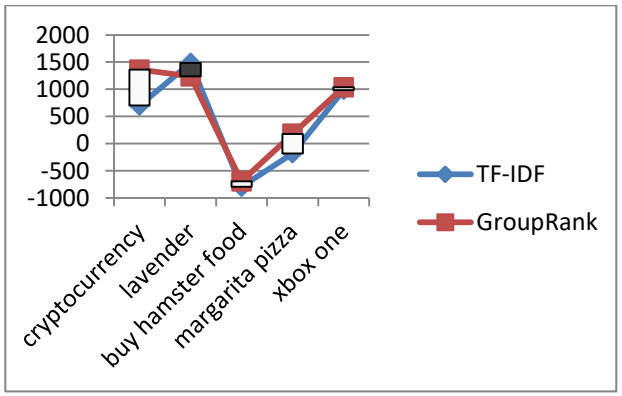

Figure 5. Grade comparison.

In summary, GroupRank works best when the top candidate results of TF-IDF are all relevant. GroupRank can be used as a spam filter or a query expansion modul, too. In our experimental results, GroupRank removed many spam groups in the search results. It also found the relevant groups having different but related words in their content.

6. Conclusion and Future Works

In this paper, we proposed a method to improve the ranking of the Telegram groups called

GroupRank. GroupRank uses the group membership records to infer the associative connection among groups. It is based on the fact that similar users prefer to join similar groups. In our experimental results, GroupRank improved the pure TF-IDF ranking by %19.

GroupRank only considers the current membership records. Our speculations show a high rate of join and leave activity among the users. Taking the membership history into account might lead to better results. Moreover, GroupRank treats all members of the group equally. Treating the administrators, active users, and inactive users differently can be a possible future work. On the other hand, GroupRank can be extended to rank the users instead of groups. For this purpose, it can be viewed as a recommender system rather than a ranking method.

We used several weights and coefficients in our algorithm. Most of them were set based on our own limited observations and possibly biased opinions over a small dataset. Creating a bigger dataset and use of machine learning for refinement of weights and coefficients is a possible future work.

Our experimental results showed that the GroupRank worked best when the top candidate TF-IDF results were all relevant. We used the top-5 TF-IDF results for associative calculations in our experiments, assuming that they were the best candidates. Sometimes there were 1 or 2 irrelevant results on the top-5 list, which had a negative effect on the quality of the final results. A better candidate selection is also a possible future work. It is possible to eliminate all those negative effects.

References

[1] A. Eugene, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information. " in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2006, pp. 19-26.

[2] B. Michael, X. Wang, D. Metzler, and M. Najork, "Learning from user interactions in personal search via attribute parameterization. "" in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ACM, 2017, pp. 791-799.

[3] E. Fredrik, P. Bródka, A. Borg, and H. Johnson, "Finding influential users in social media using association rule learning. " *Entropy* vol. 18, no. 5, pp. 164-179, 2016.

[4] M. Sandy, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data. " in *Big Data international conference*, 2013, pp. 111-118.

[5] J. Thorsten, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback. " in *ACM SIGIR Forum*, ACM, 2017, pp. 4-11.

[6] R. Stephen, and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond. " *Foundations and Trends® in Information Retrieval* vol. 3, no. 4, pp. 333-389, 2009.

[7] S. Gerard, A. Wong, and C. Yang, "A vector space model for automatic indexing. " *Communications of the ACM* vol. 18, no. 11, pp. 613-620, 1975.

[8] W. Jianshu, E. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers. " in *Proceedings of the third ACM international conference on Web search and data mining*, ACM, 2010, pp. 261-270.

[9] Y. Yuto, T. Takahashi, T. Amagasa, and H. Kitagawa, "Turank: Twitter user ranking based on user-tweet graph analysis. " in *International Conference on Web Information Systems Engineering*, Springer, Berlin, Heidelberg, 2010, pp. 240-253.

[10] Y. Ming, J. Sang, and C. Xu, "Unified youtube video recommendation via cross-network collaboration. " in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ACM, 2015, pp. 19-26.

[11] X. Chonghuan, "A novel recommendation method based on social network using matrix factorization technique. " *Information Processing & Management*, vol. 54, no. 3, pp. 463-474, 2018.

[12] A. Sajad, M. Meghdadi, and M, Afsharchi. "A social recommendation method based on an adaptive neighbor selection mechanism. " *Information Processing & Management* vol. 54, no. 4, pp. 707-725, 2018.

[13] G. D. Feltoni *et al*. "Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization. " *Future Generation Computer Systems* vol. 78, pp. 430-439, 2018.

[14] Z. Hong, D. Ge, and S. Zhang, "Hybrid recommendation system based on semantic interest

community and trusted neighbors. ” *Multimedia Tools and Applications* vol. 77, no. 4, pp. 4187-4202, 2018.

[15] B. Thirunavukarasu, R. Nayak, and C. Yuen, “People to people recommendation using coupled nonnegative Boolean matrix factorization. ” in *IEEE International Conference on Soft-Computing and Network Security*, Coimbatore, India, 2018, pp. 14-16.

[16] Z. Fattane, M. Kahani, and E. Bagheri, “Mining user interests over active topics on social networks. ” *Information Processing & Management* vol. 54, no. 2, pp. 339-357, 2018.

[17] C. L. A. Gonzalez, and S. N. Alves-Souza, “Social network data to alleviate cold-start in recommender system: A systematic review. ” *Information Processing & Management* vol. 54, no. 4, pp. 529-544, 2018.

[18] L. Preethi, G. D. F. Morales, and A. Gionis, “Finding topical experts in Twitter via query-dependent personalized PageRank. ” in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ACM, 2017, pp. 1-19.

[19] Q. Z. Tony, C. Zhuo, W. Tan, J. Xie, and J. Ye, “Large-Scale Targeted Marketing by Supervised PageRank with Seeds. ” in *International Conference on Machine Learning and Data Mining in Pattern Recognition*, Springer, Cham, 2018, pp. 409-424.

[20] H. Richang, C. He, Y. Ge, M. Wang, and X. Wu, “User vitality ranking and prediction in social networking services: A dynamic network perspective. ” *IEEE Transactions on Knowledge and Data Engineering* vol. 29, no. 6, pp. 1343-1356, 2017.

[21] A. Majed, M. AlQurishi, M. AlRakhami, M. M. Hassan, and A. Alamri. “Reputation-based credibility analysis of Twitter social network users. ” *Concurrency and Computation: Practice and Experience* vol. 29, no. 7, pp. 4676–4681, 2017.

[22] L. S. Ling, R. Chiong, and D. Cornforth, “Ranking of high-value social audiences on Twitter. ” *Decision Support Systems* vol. 85, pp. 34-48, 2016.

[23] K. Miltiadis, L. Mitrou, V. Stavrou, and D. Gritzalis, “Profiling online social networks users: an omniopticon tool. ” *International Journal of Social Network Mining* vol. 2, no. 4, pp. 293-313, 2017.

[24] B. Ricardo, “Predicting user behavior in electronic markets based on personality-mining in large online social networks. ” *Electronic Markets* vol. 27, no. 3, pp. 247-265, 2017.

[25] M. S. Ahmad, M. Jalali, N. Misaghian, S. Shamshirband, and M. H. Anisi. “Community detection in social networks using user frequent pattern mining. ” *Knowledge and Information Systems* vol. 51, no. 1, pp. 159-186, 2017.

[26] T. A. Kumar, F. Zarrinkalam, and E. Bagheri, “Topic-Association Mining for User Interest Detection. ” in *European Conference on Information Retrieval*, Springer, Cham, 2018, pp. 665-671.

[27] H. Jiangning, H. Liu, R. Y. K. Lau, and J. He, “Relationship identification across heterogeneous online social networks. ” *Computational Intelligence* vol. 33, no. 3, pp. 448-477, 2017.

[28] V. Nobahari, M. Jalali, and S. J. S. Mahdavi, “ISoTrustSeq: a social recommender system based on implicit interest, trust and sequential behaviors of users using matrix factorization. ” *Journal of Intelligent Information Systems* vol. 1, pp. 1-30, 2018.

[29] V. Faridani, M. Jalali, and M. V. Jahan, “Collaborative filtering-based recommender systems by effective trust. ” *International Journal of Data Science and Analytics* vol. 3, no. 4, pp. 297-307, 2017.