AlMahmoud and AlKhalifa *J Big Data* (2018) 5:39
https://doi.org/10.1186/s40537-018-0147-2

Journal of Big Data

RESEARCH Open Access

# TSim: a system for discovering similar users on Twitter

Hind AlMahmoud* and Shurug AlKhalifa

*Correspondence: hindSalmahmoud@gmail.com
Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

**Abstract**

This paper presents a framework for discovering similar users on Twitter that can be used in profiling users for social, recruitment and security reasons. The framework contains a novel formula that calculates the similarity between users on Twitter by using seven different signals (features). The signals are followings and followers, mention, retweet, favorite, common hashtag, common interests, and profile similarity. The proposed framework is scalable and can handle big data because it is implemented using the MapReduce paradigm. It is also adjustable since the weight and contribution of each signal in calculating the final similarity score is determined by the user based on their needs. The accuracy of the system was evaluated through human judges and by comparing the system's results against Twitter's Who To Follow service. The results show moderately accurate results.

**Keywords:** Twitter, MapReduce, Similarity on social media, Big data

## Introduction

Lately, social networks have become a vital part of our lives. Among many different uses, most people use social networks to communicate and stay informed. Twitter, a microblogging site, is currently one of the most popular social network sites. Users follow different accounts such as friends, celebrities, or companies to get information through 280 character messages (or tweets). There are currently 1.3 billion registered users on Twitter with 330 million of them active users generating 500 million tweets daily [1]. Analyzing Twitter content has recently gained a lot of attention due to its popularity all over the world and the significance of its content in detecting patterns and inferring hidden information.

A significant portion of analyzing Twitter content goes into analyzing the opinions and behavior of its users. Detecting similarity between users based on their produced content, behavior, interests, and activities is an important application of Twitter content analysis as could be seen in [2–4]. Detecting similarity could be used in profiling users for security, recruitment and social reasons. Governments could use it to identify persons who impose a threat to the security of their people by identifying one individual and finding others similar to her/him. Businesses could benefit from it in recruitment and target marketing by identifying candidate Twitter users and finding similar ones to them. Individuals could use it to find similar users to them or to others who they are interested in.

Most of the previous analysis of Twitter user behavior and similarity detection has focused on individual users and their social followers' graph, in which the nodes for two users are connected if one follows the other. In contrast, we propose a framework to generate a list of similar users starting from a single user based on different parameters, not just the follow relationship. The proposed system, TSim, enables users to input a single Twitter user ID, and get in return a list of users similar to her/him. TSim computes similarity between users using a novel similarity formula that is both comprehensive and flexible. To process the largest possible number of users, TSim uses the MapReduce model. MapReduce is a distributed programming model that was proposed by [5] and is used to analyze a great number of user accounts in order to generate "similar" users. The concept of "similarity" is a very subjective matter. Each person might view what it means differently. Hence, in TSim, we use a formula to compute similarity that includes seven different signals such as common interests and who they are following. To allow for different definitions of similarity, the user can change the weight associated with each of these seven signals to better serve their definition of similarity. In terms of evaluating the accuracy of the proposed formula, two methods were used to measure the similarity formula's accuracy. The first method is using human judges to evaluate the results of the system. The other method is comparing our results against Twitter's Who To Follow (WTF) [6] results. The "Evaluation" section of this paper contains further details on this.

The rest of this paper is organized as follows, we start by reviewing the latest work related to the system proposed here. We follow that with a thorough description of the formula used in determining similarity and the objective of each of its seven parts. Afterwards, we describe the architecture of TSim along with the particulars of implementing each part of the similarity formula. We conclude the paper with an evaluation of the results produced by TSim and a discussion of the issues related to accuracy.

## Related work

Twitter's own researchers share their proposed framework for finding similar Twitter accounts in [2], the aim of their work is to discover similar users on Twitter using different signals. Their framework is implemented using Pig on Hadoop and machine learning mechanisms to discover similar users in Twitter. It is highly scalable and can find similar accounts for hundreds of millions of users on daily basis. There are many signals that can be used to identify the similarity between users on Twitter. However, in [2] they found that only four signals are useful, which are cosine follow score, number of suggestions' followers, page rank score, and historic follow-through rate. Some signals were explored but they did not find them useful such as mutual follow, common topics of interests, location, and email domain. Their framework consists of three components: candidate generation, model learning, and regression.

As mentioned in [2] the similarity computation takes interactions into account by running the similarity computation (cosine similarity) on RealGraph. RealGraph was proposed in [7] where the authors at Twitter share a framework for predicting the user interaction named RealGraph. RealGraph consumes heterogeneous interaction data to effectively predict potential user interaction in the future. The prediction score of the user interaction can also be interpreted as connection strength, which enables a diverse set of applications to use the RealGraph such as discovering similar users. RealGraph is

used to compute relationship strength for ties based on directed interactions between users. It produces a directed and weighted graph where the nodes are Twitter users, and the edges are labelled with interactions between a directed pair of users. Furthermore, each directed edge also has a weight that is the probability of any interaction going from the edge source to the edge destination in the future. The framework learns a logistic regression based model using historical data and then scores the edge features using the model to produce the weight [7]. The first application of RealGraph in Twitter was the Who To Follow (WTF) feature [6]. WTF is a Twitter's user recommendation service, which is responsible for creating millions of connections daily between users based on shared interests and common connections. The authors at Twitter share the architecture of WTF in [6].

The similarity between users on Twitter is mainly based on the similarity between the users' signals. Signals could represent produced content (tweets), behavior, personal profile, or a combination of these features. In [3] the authors proposed a methodology for discovering similar users on Twitter based on the similarity of the produced content between users. Their similarity metric was based on the comparison coefficients of hashtags, mentions, URLs, and the domains of those URLs that an account has included in its tweets. Their similarity metric (SM) was calculated as the combination the four coefficients and the three factors. They evaluated their similarity metric based on the user ratings for the results obtained from the Similarity Metric. Their results show that their similarity metric works efficiently enough according to the evaluators' opinion.

Similarity between users on Twitter could help in finding groups of similar users. In [4], they proposed a methodology for identifying user communities on Twitter, by defining a number of similarity metrics based on their shared content, following relationships and interactions. They used common followers and friends, hashtags, reply and user mention similarities to calculate the similarity between two users. They calculated the following relationship similarity by examining the following relationship between the two users. To compute the hashtag similarity, they listed all the hashtags occurring in the tweets of a user to form a single document, then they computed the tf-idf weights of the vector space model representation of the hashtags. The users' reply similarity is calculated as the frequency of replies between two users, and the number of users that both users reply to. The user mention similarity is computed as the number of times a user mentioned another user. Finally, the Total User Similarity is calculated as a linear combination of all the individual signal similarity measures. Moreover, they proposed a new methodology based on latent Dirichlet allocation to extract user clusters discussing interesting local topics and eliminate trivial topics. They evaluated their methodology on a group of users interested in programming.

This section reviewed the work of similarity in Twitter which is directly relevant to our project. Discovering similar users on Twitter has been done by [2–4]. Below, we have highlighted the differences between our system and the work in [2–4].

Although the output of [2] is similar to our output, our work differs from it mainly due to the following reasons:

1. The main goal of our work is to develop a framework that can be used by individuals, companies, and governments for user profiling starting from a single user. In con-

trast, Twitter developed a service which is embedded in Twitter to help in providing link recommendations, accurate advertisement targeting, and enhance user experience as stated in their paper.
2. Unlike Twitter's similarity formula, our similarity formula is comprehensive and flexible since it allows users to dynamically change the weights of the used signals according to their preferences.
3. Our work uses MapReduce paradigm heavily in processing the different signals used in determining the similarity level between the user accounts. On the other hand, [2] applies graph algorithms to the twitter user interaction graph described in [7] in determining similar users.

Also, the work in [3] seems similar to our work. However, our work differs in the following:

1. Our similarity formula is flexible since it allows users to dynamically change the weight of the used signals according to their preferences while the similarity metric in [3] is fixed.
2. Their similarity metric focuses on the content signals only (e.g. hashtag, mention) since their definition of similarity is the content similarity. However, our similarity formula is comprehensive and it takes into account the relationships signals such as the common followings/followers.

The work in [4] is similar to our work since we both use a variety of Twitter signals. However, our work uses MapReduce in processing the different signals used in determining the similarity levels between the user accounts, which gives the proposed formula high scalability. Moreover, they multiply each signal similarity with a parameter to produce a number of possible partitions, or clusters of users, while in our work we give the users the flexibility to give importance levels to the similarity signals through the weight parameter. Also, the aim of their work is to find clusters of users based on their interests while our work aims to profile users.

## Proposed similarity formula

This section will abstractly describe the formula used to compute the similarity score between two users: one is our examined user (input user) and the other is a candidate user. As will be shown next, this formula is a simple weighted summation of the scores of the seven signals used in determining the similarity in each characteristic (signal) of the two users. In Table 1, we show the total similarity formula and the different parts of it. We follow that with a description of each similarity signal that feeds into the total similarity formula. Due to the large number of processed data, this formula will not be computed as is. It will be transformed into a parallel processing framework through the MapReduce programming model. The next section describes the proposed framework in greater detail.

**Table 1 Total similarity formula and the formulas used for each signal**

**The total similarity formula**

$$Sim_{\text{Total}}(u_i, u_j) = \sum_{m=1}^{7} \left( Sim_m(u_i, u_j) * \text{weight}_m \right)$$

where, $u_i$ is the examined user—*TSim* attempts to find similar users to it
$u_j$ is the candidate user—the user that *TSim is* computing its similarity to $u_i$
$\text{Sim}_m$ is the score of signal m similarity between $u_i$ *and* $u_j$. $Sim_1$ through $Sim_7$ explained below
$weight_m$ is the weight assigned to signal *m* score

| Name | | Formula | Explanation |
|---|---|---|---|
| Signal 1 or $\text{Sim}_1$ | Followings and Followers Relationship Similarity | $\text{Sim}_{\text{Relationship}}(u_i, u_j) = \begin{cases} 1 \text{ if the candidate user appears in one list} \\ 2 \text{ if the candidateuser appears in two lists} \\ . \\ . \\ . \\ n + k \text{ if the candidate user appears in all lists} \end{cases}$ | **n** is the number of the $u_i$'s followers<br>**k** is the number of the $u_i$'s friends |
| Signal 2 or $\text{Sim}_2$ | Mention Similarity | $\text{Sim}_{\text{Mention}}(u_i, u_j) = \sum_{l=1}^{w} \frac{\text{twtsThrd}(l, u_i, u_j)}{\text{twtsThrdTot}(l, u_i)} * \frac{1}{\text{accntsTwt}(l, u_i)}$ | **twtsThrd** is a function that returns the number of $u_i$ tweets in the communication thread l with $u_j$ that mention the account $u_j$<br>**twtsThrdTot** is a function that returns the total number of tweets in the communication thread l.<br>**accntsTwt** is the total number of accounts in the tweets in thread l<br>**w** is the total number of communication threads mentioning both $u_i$ and $u_j$ |
| Signal 3 or $\text{Sim}_3$ | Retweet Similarity | $\text{Sim}_{\text{Retweet}}(u_i, u_j) = \text{numOfTwtsInRetwtList}(u_i, u_j)$ | **numOfTwtsInRetwtList** is the number of $u_j$ tweets that $u_i$ retweeted |
| Signal 4 or $\text{Sim}_4$ | Favorite Similarity | $\text{Sim}_{\text{Favorite}}(u_i, u_j) = \text{numOfTwtsInFavList}(u_j, u_i)$ | **numOfTwtsInFavList** is the number of $u_j$ tweets that $u_i$ favorited |
| Signal 5 or $\text{Sim}_5$ | Common Hashtags Similarity | $\text{Sim}_{\text{Hashtag}}(u_i, u_j) = \sum_{l=1}^{w} \frac{1}{1 + HTOffset(u_i, u_j, HT_l)}$<br>where,<br>$HTOffset(u_i, u_j, HT) = \lvert \text{PT}(u_i, \text{HT}) - \text{PT}(u_j, \text{HT}) \rvert + \lvert \text{NT}(u_i, \text{HT}) - \text{NT}(u_j, \text{HT}) \rvert + \lvert \text{NTT}(u_i, \text{HT}) - \text{NTT}(u_j, \text{HT}) \rvert$ | **PT** is a function that takes in a user id and a hashtag HT and returns the number of positive tweets of the user in the hashtag<br>**NT** is a function that takes in a user id and a hashtag HT and returns the number of negative tweets of the user in the hashtag<br>**NTT** is a function that takes in a user id and a hashtag HT and returns the number of neutral tweets of the user in the hashtag<br>**w** is the total number of hashtags that both $u_i$ and $u_j$ tweeted in |

**Table 1 (continued)**

| Name | | Formula | Explanation |
|---|---|---|---|
| Signal 6 or $Sim_6$ | Common Interests Similarity | $Sim_{Interests}\left(u_i, u_j\right) = \text{count}\left(\text{ints}(u_i) \cap \text{ints}\left(u_j\right)\right)$ | **Ints** is a function that takes in a user id and returns his/her top 5 interests after performing topic analysis to his/her tweets |
| Signal 7 or $Sim_7$ | Profile Similarity | $Sim_{Profile} = \left[\text{gender}(u_i) \textit{is equal to } \text{gender}\left(u_j\right)\right] + \left[\text{language}(u_i) \textit{is equal to } \text{language}\left(u_j\right)\right] + \left[\text{location}(u_i) \textit{is equal to } \text{location}\left(u_j\right)\right]$ | **Gender** is a function that takes in a user id and returns its gender from the user's profile on Twitter<br>**Language** is a function that takes in a user id and returns its language from the user's profile<br>**Location** is a function that takes in a user id and returns its location from the user's profile |

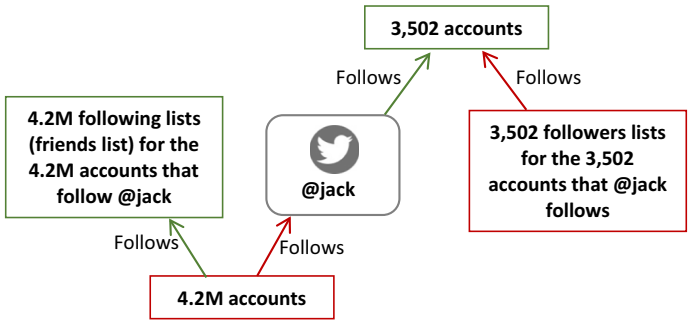


**Fig. 1** Lists examined when scoring the follower/following relationship between two users

#### Signal 1: Followings and followers relationship similarity

The first characteristic that comes to mind when deciding how two users are similar in Twitter is by examining their following relationship and the common users between them. We assume similar people tend to follow each other. Furthermore, when a person follows some account, we consider other followers of this account similar to our user. Also, when the followers of our user follow another account, we will consider this account similar. Each appearance of an account in the lists of followers to the account that our user follows scores a point for this account. Likewise, each appearance of an account on the follow lists of the followers of the examined account scores a point to this account. The more lists the account appears in, the higher the score. By letting n be the examined user's followers and k is the examined user's friends, there will be $n+k$ different lists, which mean $n+k$ score.

If we take for example the account of Twitter founder Jack Dorsey @jack. He has 4.2 M followers and 3502 followings—meaning that he follows 3502 accounts on Twitter. The relationship similarity is calculated for each of the 4.2 M followers, 3502 followings, the followings' list of each one of the 4.2 M followers which results in 4.2 M different lists, and the followers' list of each one of the 3502 followings which results in 3502 different lists. If a user appears in the following list, it means he/she might appear in all other users' followers lists except himself, so $n-1$. If a user appears in the followers list, it means he/she might appear in all other users' followings lists except himself, so k-1. Since the user might appear in all of the examined user's followings and followers list so $(k-1)+(n-1)+2=n+k$. All users in all of these lists will be examined and scored. Signal 1 row in Table 1 shows the actual formula used to compute the score of each of these examined users. Most of them will probably score 1. Figure 1 demonstrates the number of lists of users to be examined and scored.

#### Signal 2: Mention similarity

Mention is one of the content signals that we consider in our similarity formula. If a user mentions another user in a tweet, it indicates that there is a relationship between them, and maybe similarity. However, the strength of mention changes based on the number of mentions and also the number of other Twitter accounts in the tweet. Every time our user mentions another user, we consider this a communication thread. The longer the communication thread, the stronger the relationship and hence, the higher the similarity

score. All examined user's tweets that contain a mention are used to calculate the mention similarity. Table 1 shows the similarity formula based on mentions. As shown in Table 1, both the number of tweets in the communication thread and the number of accounts in the conversation are taken into account. For instance, if the examined user A and candidate user B are chatting in a thread where the total number of tweets is six. User A appeared in five tweets while user B appeared in four tweets. There were also two additional users named C and D who appeared in all four tweets that B appeared in. According to the similarity formula, it is calculated as: (5/6) * (1/3) = 0.278 for each tweet. Then afterwards, all the threads that B appeared will have their scores computed and summed up to produce the final signal 2 score for user B.

#### Signal 3: Retweet similarity

In the similarity formula, the retweet signal is used because if a user retweets a tweet, he/she distributes the tweet because there is something that is implicitly interesting in that tweet. Therefore, the examined user and the original user of the tweet have something in common. The retweets similarity score is calculated according to the formula in Table 1, which is a point for each retweet.

#### Signal 4: Favorite similarity

We also consider what tweets the user favorites as one of the content signals in our similarity formula. If a user favors another user's tweet it means he/she likes the content of that tweet, and this tells us that these users have something in common and they could be similar. The favorite similarity score is calculated according to the formula in Table 1, which is a point for each tweet favorited by the examined user.

#### Signal 5: Common hashtags similarity

Common hashtags are one of the content signals in the similarity formula. If a user writes in a specific hashtag, it implies that she/he is interested in the issue or idea of that hashtag. However, if two users write in a specific hashtag it does not necessary mean they are similar because they might have conflicting opinions. To overcome this issue, the sentiment analysis of the tweets written by the user in this hashtag (negative, positive, neutral) is considered. Hashtag similarity is calculated as shown in Table 1. The sentiment is calculated using both Stanford's Core NLP [8], which gives each piece of text a sentiment score, and the sentiment score for emoji icons. For Stanford's Core NLP, it classifies the tweet text into seven sentiment classes, from the very negative to very positive. Therefore, the possible values are $-3$, $-2$, $-1$, 0, 1, 2, and 3 for each tweet. To increase the accuracy of calculating tweets' sentiment, the sentiment score from Stanford is combined with emoji icons' sentiment to produce the final sentiment score of the tweet. Emoji icons are considered in sentiment analysis of tweets since they have a strong relationship with emotions and are used heavily on Twitter [9]. To calculate the sentiment of the emoji part, emoji icons are classified into positive, negative, and neutral. Most of the emoji's have a clear meaning for the emotions they represent. However, for those emoji's that were hard to classify, 146 people were surveyed on what they felt the sentiment is for each of these emoji's and their responses were used to assign a score

for each of the sentiment-ambiguous emoji's. In the formula in Table 1, the tweets of both users are examined for sentiment in each hashtag they both tweeted in. The closer their sentiment in the hashtags, the higher the score for the candidate user with a maximum of 1 for each hashtag.

#### Signal 6: Common interests similarity

Common interests are one of the content signals in the similarity formula. Two users are considered similar to each other if they share the same interests. Users' interests are extracted from user's tweets, we created a customized simplified lexicon between interests and the words that usually come with these interests. This lexicon was created using an online English dictionary [10] that has subjects and sub-subjects. From that dictionary, we extracted only the subjects that indicate interests such as Art, Camping, Cooking, Computer and Technology, Election and Politics, and Sports. The user tweets are analyzed for the words connected to the interests and subsequently, the highest five interests are determined. Similarity score for this signal is calculated according to the formula in Table 1, which basically intersects the interests of the two users. The highest possible score would be 5.

#### Signal 7: Profile similarity

The profile similarity score consists of three parts; gender, language, and location. Since this information is private in Twitter, the gender of the examined user and candidate user are determined based on their names. For gender detection, a dictionary is created by collecting all names of babies from 1880 to 2015 from the Social Security Administration SSA [11]. SSA website allows you to download the baby names as text files. These files contain the name, the gender of the name, and the number of babies born that year who hold that name. For instance, Emma, F, 20,355. If a name appears with two different genders, we check the number of babies' names for each gender. If the number of female babies was more than male babies, then we associate this name with female and vice versa. In the end, we will have a long list of unique names and their gender. The location and language are optional parameters in the Twitter profile. They are compared for the examined user against the candidate user if they exist in both. The similarity formula gives one point for each matching parameter. Making the similarity score for this signal no more than 3. Table 1 shows the formula used to compute the score of this signal.

#### Signals normalization

We have performed several experiments to reach a reasonable state where all signals have almost a similar effect on the final score. We did that to ensure that the user-defined weights alone decide the importance of a signal over another. For some signals, it was straightforward such as favorite and retweet. Each retweet or favorite equals one point. For mention, the score we got from that signal was reasonable, so we did not need to do any normalization. Followings and followers was the hardest signal to normalize, because the output scores of this signal varies from tens to hundreds. The best normalization we reached was by multiplying the score by ten, then divide it by $n+k$, where n is the examined user's followers and k is the examined

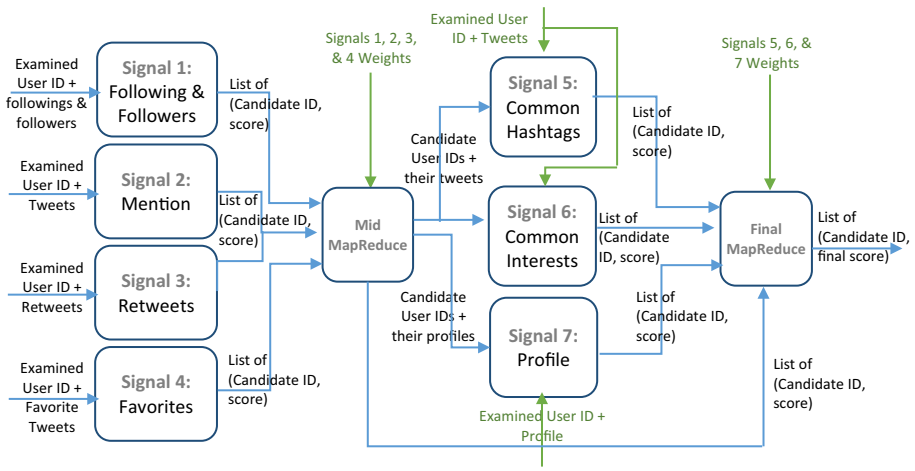


**Fig. 2** A graph that shows the overall architecture of TSim and how the nine different MapReduce jobs fit together

user's friends. For the hashtag, the normalization is done for the sentiment part, based on sentiment score from both Stanford and the emoji sentiment. The sentiment scores are mapped into three scores only, which are 1, 2, and 3.

There is no normalization for interests or profile signals because the score of interests' signal is from 0 to 5 based on the common interests between them. Similarly, the score of profile signal is from 0 to 3 based on the common profile elements between them.

### Signal weight

A question that may occur to the user of TSim is how to set the weights of each of the signals. As we mentioned earlier, the concept of "similarity" is very subjective. To allow each user to get results based on his/her definition of similarity, the user is expected to understand the meaning of each signal and how the score is computed for this signal. Based on this knowledge and understanding, the user is expected to set the weights according to their definition of similarity and their purpose for finding similar users. We also expect the user to apply the weights and view the results through TSim interactively. So the user is actively trying out different weights, judging the results, adjusting the weights, and trying again until he/she is satisfied with the returned results.

To assist the user in deciding the initial weights, a preset group of configurations, each with its purpose and a its own set of weights to serve that purpose, will be available to the user. He/she could then adjust these weights or start from scratch. Obviously the user could choose not to incorporate any weights at all and just set them all to 1, which will give each of the signals an equal weight.

## TSim architecture

To implement the proposed similarity formula, we need to analyze a large number of Twitter users. TSim consists of nine MapReduce jobs organized into two phases as shown in Fig. 2. MapReduce is a distributed programming model that was proposed by

[5] and is used in TSim to analyze the great number of user accounts in order to generate similar users. The two phases include two different sets of signals. The first phase consists of four signals; following and followers, mention, retweet, and favorite. This phase is followed by a MapReduce job that sorts and filters and results and passes them to the three remaining signals; common interests, hashtag, and profile. The output of these signals is passed to final MapReduce job that sorts and filters the final results.

The first phase consists of four signals, followings and followers, retweet, favorite, and mention. These were chosen as a first phase because they have a clear user collection starting point. Whereas the rest of the signals have to find a starting point. For example, the common interests signal should find users with interests similar to our examined user. Extracting the interests of our user and then finding random users on twitter that share his/her interests would return a huge number of users with most of them being irrelevant and would score zero on the rest of the signals. So instead of exploring the large space of Twitter users at random, we apply the signal formula on the users who actually were returned from the first phase. In the first phase, the signal has a clear starting point. For example, in the favorite signal, we examine only the users who our user has favorited their tweets, which is a limited number of users. After phase one runs the four signals in parallel, we include a middle MapReduce module. This module will sum up all scores, along with their user-defined weights, of the previous four signals and filter users based on their scores. After that, the users who passed that middle MapReduce module will go through the remaining three signals in parallel: interests, hashtag, and profile similarity. We will retrieve the tweets of users to extract common interests and hashtags. We will also retrieve their profiles to compare language, location, and gender with the input user. Finally, the output of these three signals will go through the final MapReduce to compute the final scores and factor in the user-defined weights of signals.

At first, we designed the core system so that all signals are running in parallel except the profile signal because it will only add a number between 0 and 3 to all candidates after their signals' similarity score was calculated. However, we discovered that this architecture has several significant issues that forced us to change it. The old architecture consists of six modules running on parallel, each will get a group of users from twitter, analyze their data, compute their scores, and pass them to the total similarity calculator module. In addition to the starting point issue discussed above that distinguished phase one signals from phase two signals, we found another problem with the old architecture. Since our scoring formula is incremental, users with higher scores will likely have gotten these scores from being processed by multiple signals. So it is better to limit the users who are considered for similarity by starting with a large group of users and then continuing to analyze them based on the different signals. If we let each signal get a different set of users based on its semantics, we would have ended up with a large number of users that have very low scores. Rarely would we have come across a user with a high score resulting from him/her being processed by several signals. Due to these issues, we rearranged the MapReduce jobs and ended up with the architecture in Fig. 2. In Table 2, we describe the Map and the Reduce functions associated with each signal.

To illustrate the necessity of using a two-phase architecture instead of one, we include a simple example using a signal from phase 1 (retweets) and another form phase 2

**Table 2 Brief descriptions of the map and the reduce functions used in processing each signal**

| MapReduce job | Map function | Reduce function |
|---|---|---|
| Signal 1: Followings and followers relationship similarity | It takes in the examined user ID and each of his/her following and followers<br>It simply produces pairs of (follower/following user id, "1") | The input will be every follower/following user ID and a list of "1"s depending on how many times this user id appeared in the different lists<br>The reduce adds up these "1"s to produce the follower/following user ID along with the sum of these ones, which is its score on this signal |
| Signal 2: Mention similarity | It takes in the examined user ID and each of his/her tweet threads<br>It extracts the user IDs in these tweets (preceded by @ symbol)<br>It calculated the score for each user ID based on the formula in Table 1<br>It outputs each user ID along with its score | The input will be every user ID mentioned in the tweets of the examined user along with a set of scores for each thread this user was mentioned in<br>The reduce adds up these scores to produce the mentioned user ID along with the sum of these scores, which is the user's score on this signal |
| Signal 3: Retweet similarity | It takes in the examined user ID and each of his/her retweets<br>It simply produces pairs of (original tweeter user id, "1") | The input will be every user ID the examined user has retweeted their tweets and a list of "1"s depending on how many times the examined user retweeted for this particular user<br>The reduce adds up these "1"s to produce the retweeted user ID along with the sum of these ones, which is its score on this signal |
| Signal 4: Favorite similarity | It takes in the examined user ID and each of his/her favorited tweets<br>It simply produces pairs of (original tweeter user id, "1") | The input will be every user ID the examined user has favorited their tweets and a list of "1"s depending on how many times the examined user favorited for this particular user<br>The reduce adds up these "1"s to produce the favorited user ID along with the sum of these ones, which is its score on this signal |
| Signal 5: Common hashtags similarity | It takes in the candidate user ID and each of his/her tweets that have the hashtag symbol (#)<br>It compares the sentiment of tweets against the sentiment of the examined user's tweets in the same hashtag (obtained in preprocessing) using the formula in Table 1. (HTOffset)<br>It produces (candidate ID, Hashtag + score) | The reduce function will receive a candidate user ID and a list of pairs of hashtags and scores<br>It will loop through this list and sum the scores with the same hashtag<br>Then it will use the similarity formula in Table 1 to compute the final score for each candidate<br>Produce candidate ID and score |
| Signal 6: Common interests similarity | It takes in the candidate user ID and a list of his/her tweets<br>Applies LDA to get the top 5 interests<br>Computes the score after comparing with the examined user's top 5 interests (obtained in preprocessing) according to the formula in Table 1<br>Produce (candidate ID, score) | The Reduce function simply takes the input and passes as output |

**Table 2 (continued)**

| MapReduce job | Map function | Reduce function |
|---|---|---|
| Signal 7: Profile similarity | It takes in the candidate user ID and his/her profile info<br>Computes the score after comparing with the examined user's gender, location and language (obtained in preprocessing) according to the formula in Table 1<br>Produce (candidate ID, score) | The Reduce function simply takes the input and passes as output |
| Mid and final MapReduce | Takes in the candidate user ID along with his/her score<br>Produces (candidate ID, signal weight + score) | The reduce function will receive a candidate user ID and a list of pairs of signal weights and scores<br>It will loop through this list and sum the scores with the same weight<br>Then it will multiply the summed up score by the associated weight and sums up the weighted sums to produce the score for that candidate<br>Produce candidate ID and score |

**Table 3 Different configurations with different weights**

| Configuration | Signals weights | | | | | | |
|---|---|---|---|---|---|---|---|
| | Followings and followers | Mention | Retweet | Favorite | Hashtag | Interests | Profile |
| Content based | 0 | 0 | 2 | 2 | 1 | 2 | 0 |
| Interaction based | 10 | 3 | 0 | 0 | 0 | 0 | 0 |
| Personality based | 0 | 0 | 1 | 0 | 0 | 3 | 2 |

(common hashtags) to demonstrate how the phase 2 signal is very difficult to use in phase 1. Retweets signal accesses the retweets of the examined user. Any other user that was retweeted by our examined user is passed to the signal and scored accordingly. With this signal, we have a limited number of users to start with. The number maybe big, but still limited and we know that our examined user found each one of them interesting enough to retweet one of their tweets. After these users are scored, they are aggregated with other users from the other phase 1 signals and their scores are summed up. Now let us consider the common hashtags signal. If we are to include it in phase 1, it means that we need to get all the users who tweeted in the hashtags that our examined user has tweeted in. If the hashtags were trends at some point, then the number of users to be examined would be in the millions. Most of these users would never appear on the other signals input, and hence would only get the hashtag score then, in subsequent phases, discarded. To avoid the unnecessary processing of millions of users who would end up irrelevant, we limit the input to the hashtag signal to the user IDs that actually passed and were scored in phase 1. These users have already shown similarity to our examined user in four signals. Therefore, it makes sense to accumulate their scores based on the three remaining signals instead of running all signals in parallel and ending up with many low-scored users.

**Table 4 Content based configuration results break-down of scores**

| User ID | Retweets | | Favorites | | Hashtag | | Interests | | Final score |
|---|---|---|---|---|---|---|---|---|---|
| | Score | Weighted score | Score | Weighted score | Score | Weighted score | Score | Weighted score | |
| X1 | 22 | 44 | 5 | 10 | 0 | 0 | 3 | 6 | 60 |
| X2 | 12 | 24 | 9 | 18 | 0 | 0 | 4 | 8 | 50 |
| X3 | 14 | 28 | 5 | 10 | 3 | 3 | 4 | 8 | 49 |
| X4 | 12 | 24 | 2 | 4 | 6 | 6 | 3 | 6 | 40 |
| X5 | 9 | 18 | 1 | 2 | 7 | 7 | 3 | 6 | 33 |
| X6 | 7 | 14 | 3 | 6 | 6 | 6 | 3 | 6 | 32 |
| X7 | 6 | 12 | 3 | 6 | 7 | 7 | 4 | 8 | 33 |
| X8 | 6 | 12 | 0 | 0 | 11 | 11 | 4 | 8 | 31 |
| ACM_CEO | 10 | 20 | 1 | 2 | 0 | 0 | 3 | 6 | 28 |
| unisouthampton | 9 | 18 | 0 | 0 | 0 | 0 | 4 | 8 | 26 |
| X9 | 7 | 14 | 0 | 0 | 3 | 3 | 3 | 6 | 23 |
| webscience-trust | 8 | 16 | 1 | 2 | 0 | 0 | 3 | 6 | 24 |
| TheOfficialACM | 6 | 12 | 1 | 2 | 0 | 0 | 4 | 8 | 22 |

**Table 5 Interaction based configuration results break-down of scores**

| User ID | Followings and followers | | Mentions | | Final score |
|---|---|---|---|---|---|
| | Score | Weighted score | Score | Weighted score | |
| X1 | 0.041 | 0.410 | 16.166 | 48.499 | 48.9 |
| X10 | 2.7668 | 27.668 | 0 | 0 | 27.6 |
| BBCBreaking | 2.4009 | 24.009 | 0 | 0 | 24.0 |
| BillGates | 2.3777 | 23.777 | 0 | 0 | 23.7 |
| X2 | 1.6244 | 16.244 | 2 | 6 | 22.2 |
| TEDTalks | 2.1742 | 21.742 | 0 | 0 | 21.7 |
| stephenfry | 2.101 | 21.010 | 0 | 0 | 21.0 |
| TheEconomist | 2.0439 | 20.439 | 0 | 0 | 20.4 |
| TechCrunch | 2.0332 | 20.332 | 0 | 0 | 20.3 |
| X11 | 2.0242 | 20.242 | 0 | 0 | 20.2 |
| royalsociety | 1.2334 | 12.334 | 2.5 | 7.5 | 19.8 |
| guardiantech | 1.9207 | 19.207 | 0 | 0 | 19.2 |
| BarackObama | 1.90467 | 19.0467 | 0 | 0 | 19.0 |

## Evaluation

Similarity between users in general is not a straightforward task, since similarity as a concept is subjective. Measuring similarity between social media users is even more challenging because the only means of judging their character is based on the content they produce online. As we have shown in the literature review, a lot of research has been conducted to measure the similarity on Twitter. Based on the reviewed literature, the accuracy of the proposed similarity formula was evaluated by human judges [2, 3]. We followed their example and evaluated the system through human judges and also by comparing our results against Who To Follow service from Twitter [6].

**Table 6 Personality based configuration results break-down of scores**

| User ID | Retweets | | Interests | | Profile | | Final score |
|---|---|---|---|---|---|---|---|
| | Score | Weighted score | Score | Weighted score | Score | Weighted score | |
| X1 | 22 | 22 | 3 | 9 | 2 | 4 | 35 |
| X2 | 12 | 12 | 4 | 12 | 2 | 4 | 28 |
| unisouthampton | 9 | 9 | 4 | 12 | 2 | 4 | 25 |
| X12 | 3 | 3 | 5 | 15 | 3 | 6 | 24 |
| websciencetrust | 8 | 8 | 3 | 9 | 3 | 6 | 23 |
| ACM_CEO | 10 | 10 | 3 | 9 | 2 | 4 | 23 |
| X8 | 6 | 6 | 4 | 12 | 2 | 4 | 22 |
| wef | 6 | 6 | 4 | 12 | 2 | 4 | 22 |
| ACM_President | 3 | 3 | 4 | 12 | 3 | 6 | 21 |
| Marthalanefox | 7 | 7 | 3 | 9 | 2 | 4 | 20 |
| royalsociety | 4 | 4 | 4 | 12 | 2 | 4 | 20 |
| X13 | 4 | 4 | 4 | 12 | 2 | 4 | 20 |
| X14 | 4 | 4 | 4 | 12 | 2 | 4 | 20 |
| X15 | 4 | 4 | 4 | 12 | 2 | 4 | 20 |

In order to evaluate the accuracy of the proposed formula, we conducted several experiments using an account of a well-known professor in the field of computing as the examined user. We omitted their name for privacy reasons. We will refer to our examined user from now on as ExU. Before starting the evaluation, we had to set the weights used in the similarity formula. We set three configurations and evaluated the similarity formula based on them.

#### Proposed configurations

We came up with three different configurations that could be used for different purposes. Table 3 shows the weight distribution of signals in the following configurations:

1. Content based configuration: consists of retweet, favorite, hashtag, and interests. This configuration could be used for recruitment.
2. Interaction based configuration: consists of followings, followers, and mention. This configuration could be used for security and fun purposes.
3. Personality based configuration: consists of profile, interests, retweet. This configuration could be used for security purposes.

#### Initial evaluation

We started by evaluating the formula and its different parts in the three different configurations for our examined user, ExU, by finding similar accounts. We fed TSim the account ID of ExU and got back the top similar users based on the different weight configurations. In Tables 4, 5, and 6, we show the top similar users and how they scored in each signal in the similarity formula. We use new identifiers to refer to the Twitter accounts of private users to insure their privacy. The Twitter IDs of organizations and public figures are left unchanged.

After a quick glance at Tables 4, 5, and 6, we can see that there is a couple of users who appear in all of the three tables. Namely, X1 and X2. These two users very likely share a close personal and/or professional relationship with ExU. We also notice the presence of a lot or organizations and institutes that are related to the field of ExU, which is computing.

Table 4 results, which are content-based, show that the common interests score did not have a big effect on the results despite the fact that it is the only signal that directly analyzes the content produced by the users to deduce the topics that they frequently tweet about. The reason for that is the limited size of the lexicon used in the topic modeling performed on the tweets. We discuss this matter furthermore in the Discussion section following "Evaluation" section. Conversely, the other three signals in Table 4, which are retweet, favorites, and hashtag, display a lot a variance in their results. But because of the weights assigned to each of these signals, we find that the retweet and favorites signals have more effect on the final score than the hashtag signal. In TSim, the user can set these weights based on their purpose of finding similar users on Twitter.

In Table 5, we can see that the mention signal is a very selective one. Since it scores IDs based on their interaction with ExU, we find that most IDs scored zero except for X1 and royalsociety. And since we already established that X1 shares a close personal or professional relationship with ExU, it is very logical that they have mentioned each other in their tweets. Another observation on Table 4, which is interaction-based, is that we can see that the results are dominated by institutes and organizations, which is not intuitive when you think of interactions. In the interaction-based configuration, the users who "communicate" with ExU should score the highest. The reason for this surprising result is that our chosen ExU is not very communicative on Twitter, and hence, scores from the followings and followers signal dominated this result. And since organizations tend to have more followers, finding common followers between ExU and these organizations is very likely, especially if their specialization is similar to ExU. A quick fix for this would be either to remove the followings and followers signal altogether (weight = 0), or to increase the weight assigned to the mention signal. In TSim, the user could interactively adjust the weights based on his/her needs and based on the returned results.

In Table 6, we introduce another signal, the profile signal. Unfortunately, like the signal of common interests, the profile signal does not show a lot of variance between

**Table 7 Surveyed users evaluation of the top 5 similar users**

| Similar user | Score | Similar user | Score | Similar user | Score | Similar user | Score |
|---|---|---|---|---|---|---|---|
| (a) 6 different signals configuration | | (b) Content-based configuration | | (c) Interaction-based configuration | | (d) Personality based configuration | |
| X1 | 4 | X1 | 4 | X1 | 4 | X10 | 5.5 |
| X3 | 5 | X2 | 0.5 | X10 | 5.5 | X2 | 0.5 |
| X2 | 0.5 | X3 | 5 | BBCBreaking | − 5.5 | unisouthampton | 3.5 |
| X16 | 6 | X4 | 6.5 | BillGates | 0 | X12 | − 1.5 |
| X17 | 6.5 | X5 | 5 | X2 | 0.5 | webscience-trust | 5 |
| Avg score | 7.5 | Avg score | 4.2 | Avg score | 0.9 | Avg score | 2.6 |

**Table 8 Comparison between top 5 similar users and WTF**

| Similar user | In WTF? | Similar user | In WTF? | Similar user | In WTF? | Similar user | In WTF? |
|---|---|---|---|---|---|---|---|
| (a) 6 different signals configuration | | (b) Content-based configuration | | (c) Interaction-based configuration | | (d) Personality based configuration | |
| X1 | No | X1 | No | X1 | No | X10 | No |
| X3 | Yes | X2 | Yes | X10 | No | X2 | Yes |
| X2 | Yes | X3 | Yes | BBCBreaking | No | unisouthampton | No |
| X16 | Yes | X4 | No | BillGates | No | X12 | No |
| X4 | No | X5 | Yes | X2 | Yes | webscience-trust | Yes |

users in the scores produced. The reason is that the range of the score is a number between 0 and 3 and it depends on whether ExU and the scored user are similar in gender, language, and location. We discuss the results of this signal in greater detail in "Discussion" section that follows "Evaluation" section.

#### Human evaluation

To measure the accuracy of the results returned by the system from a human point of view, we developed a survey that asked people to evaluate the similarity (very similar, similar somehow, not similar) of each of the results returned by our system to our examined user, ExU. Seven people participated in that study. Table 7 shows their evaluation of the top 5 results returned by our system using the three proposed configurations in addition to a forth one that uses six different signals with equal weights.

The evaluation of the human judges of our similarity results is promising, with the configuration using 6 signals returning the best results (Table 7a). But we have to strongly emphasize that the concept of similarity is very subjective. So some of the surveyed users assumed that similarity meant that both ExU and returned users are public figures. Other users assumed that the similarity of account type (personal or organization) was what decided similarity despite the fact that if an organization account and a personal account produced similar content and showed similar online behavior, they should be considered similar regardless of the account type. We also believe that some of the differences between users' evaluation and the returned results is due to the effect of some of the signals and weights assigned in the initial configurations. We discuss the issues concerning these signals in "Discussion" section following "Evaluation" section.

Although the results of human evaluation of the similarity were mostly encouraging, we strongly feel that human judges are difficult to rely on when measuring the accuracy of the system. Therefore, we discuss next an alternative approach to measuring the accuracy of our system.

#### "Who to Follow" comparison

To evaluate the system from another, more reliable point of view, we compared our results to the only available service that shows suggested similar users on Twitter. This service is provided by Twitter and is called Who To Follow (WTF) [6].

In order to collect similar users to our examined user, we retrieved the accounts displayed in "Who To Follow comparison" section. We collected the thirty accounts produced by the service on three different times. The results from these three times were intersected with the results of our system. Table 8 shows the similarity results compared to WTF. We had to get the accounts on WTF three times because each time you check it; you are likely to get different results. The reason for this is that WTF, according to their published paper [6], when producing similar accounts to a target account, they randomly sample the connections of this account with other accounts. This element of randomness, although returns acceptable results when thinking of following an account, is not consistent. We believe this is one of the main reasons why our results are different than the results produced by WTF.

## Discussion

A natural question that faced us is: given a user ID, have we found his/her most similar users on Twitter? Our formula proved that it discovers a list of very similar users to a given user. However, Twitter is a huge graph, and it consists of a billion users. Therefore, we cannot claim that the formula discovered all similar users to a given user, because if for instance two users are similar to each other but there is no interaction between them, it will be quite hard to find these similar users. But we believe that, based on the criteria discussed in the similarity formula section, TSim does find the most similar users.

From the results shown above, we can deduce that some signals require additional work in terms of effect on the final score. For example, the common interests signal is computed in a way that might not truly distinguish between different users. A solution to that is to expand the lexicon used in extracting the interests to include a third and maybe fourth level of sub-topics. This will exclude users who share general interests with ExU and include users who truly share the most specific of interests. Obviously, the danger of going too specific in terms of common interests is not finding enough users that share ExU's interests, and hence rely on the scores of other signals to get similar users.

Another signal that suffers from the same problem as the signal of common interests is the profile signal. This signal returns a number between 0 and 3 that basically measures whether or not the two users have common gender, language, or location with 0 meaning having nothing in common, and 3 meaning that all three attributes are common between the two users. This small range causes the score to vary a little between top users. To remedy that, an additional biography analysis could be performed to deduce the similarity of content and of style between the profile of two users. Another addition could be any common affiliation between the two users whether it is mentioned explicitly in their Twitter bio or inferred from their published contact information. We could also work more on how the location is decided. Currently, we just compare the two locations. But users tend to either overgeneralize or over-specify their location. For example, if a user specifies a location "Paris" and another user specifies location "Latin Quarter", TSim should deduce that one of them is geographically part of the other and hence these two should match on the location attribute, or at lease score higher than zero.

Another observation from the results above is the varying scale of the returned score for each of the signals. We can see that the retweets and the mentions return scores that

are an order of magnitude bigger than the rest of the signals. This should be studied and adjusted.

## Conclusion

In this paper we described TSim, a system for finding similar users on Twitter. This system takes in a single user ID then finds similar users to her/him based on a novel formula that is both comprehensive and flexible. This similarity formula uses seven different signals: followings and followers, mention, retweet, favorite, common hashtags, common interests, and profile. It allows the user to specify weights for different signals based on his/her needs. To allow the processing of a big amount of data, TSim is built using the MapReduce distributed programming model. The system was thoroughly evaluated by human judges and by comparing the results to the Who To Follow (WTF) service provided by Twitter. The results produced by TSim were promising and reasonably accurate.

To the best of our knowledge, no previous research proposed a framework similar to ours. The novelty of this project is proposing a scalable and flexible framework for finding similar Twitter accounts based on user definition of similarity and preferences that can handle huge amount of data. It is scalable because it is implemented using MapReduce paradigm, which can handle large amounts of data. It is flexible because it allows the user to manipulate the weights dynamically according to his/her preferences.

This work could be further improved by adding more signals to the similarity formula. Signals such as account type (business, government, or personal) and content analysis of the bio could greatly improve the accuracy of the similarity formula. Also, additional work could go into normalizing the scores of the different signals to allow an easier task for the system when adjusting the weights.

**Abbreviation**
ExU: the examined user.

**Authors' contributions**
The idea and the work was proposed by both of the authors. Both authors read and approved the final manuscript.

**Acknowledgements**
Not applicable.

**Competing interests**
The authors declare that they have no competing interests.

**Availability of data and materials**
Currently the data and the code is not publicly available.

**Funding**
This research project was supported by a grant from the Research Center of the Female Scientific and Medical Colleges, Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia.

## Publisher's Note



Received: 17 July 2018 Accepted: 4 October 2018
Published online: 20 October 2018

**References**

1. Smith C. 170+ Amazing Twitter statistics, *DMR*. 2016. http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/. Accessed 11 May 2016.
2. Goel A, Sharma A, Wang D, Yin Z. Discovering similar users on Twitter. In: Workshop on mining and learning with graphs, Chicago, USA. 2013.
3. Razis G, Anagnostopoulos I. Discovering similar Twitter accounts using semantics. Eng Appl Artif Intell. 2016;51:37–49.
4. Vathi E, Siolas G, Stafylopatis A. Mining interesting topics in Twitter communities. In: Computational collective intelligence. Berlin: Springer; 2015. p. 123–32.
5. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Commun ACM. 2008;51(1):107–13.
6. Gupta P, Goel A, Lin J, Sharma A, Wang D, Zadeh R. WTF: the Who To Follow service at Twitter. In: Proceedings of the 22nd international conference on world wide web, New York, NY, USA. 2013. p. 505–14.
7. Kamath K, Sharma A, Wang D, Yin Z. RealGraph: user interaction prediction at Twitter. In: Presented at the user engagement optimization workshop@ KDD. 2014.
8. Socher R, et al. Recursive deep models for semantic compositionality over a sentiment treebank, vol. 1631. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013. p. 1642.
9. Psychology Today. Health, Help, Happiness + Find a Therapist. Psychol Today. https://www.psychologytoday.com/blog-posts. Accessed 30 Dec 2017.
10. Word Lists by Theme. Wordbanks—EnchantedLearning.com. http://www.enchantedlearning.com/wordlist/. Accessed 15 Jan 2017.
11. The United States Social Security Administration. https://www.ssa.gov/. Accessed 4 Jan 2017.