Virginia Journal of Science
Volume 67, Issue 3 & 4
Fall & Winter 2016
doi: 10.25778/WHAA-RK11

Content Mining Techniques for Detecting Cyberbullying in Social Media

Shawniece L. Parker and Yen-Hung Hu

Department of Computer Science, Norfolk State University, Norfolk, Virginia 23504

ABSTRACT

The use of social media has become an increasingly popular trend, and it is most favorite amongst teenagers. A major problem concerning teens using social media is that they are often unaware of the dangers involved when using these media. Also, teenagers are more inclined to misuse social media because they are often unaware of the privacy rights associated with the use of that particular media, or the rights of the other users. As a result, cyberbullying cases have a steady rise in recent years and have gone undiscovered, or are not discovered until serious harm has been caused to the victims. This study aims to create an effective algorithm that can be used to detect cyberbullying in social media using content mining. Bullies may not use only one social media to victimize other users. Therefore, the proposed algorithm must detect whether or not a user is victimizing someone through one or more social media accounts, then determine which social media accounts are being used to carry out the victimization. To achieve this goal, the algorithm will collect information from content shared by the users in all of their social media accounts, then will determine which content to extract based on a big data technology involving phrases or words that might be used by cyberbullies. Any extracted data will reveal some insight into whether or not cyberbullying is occurring and trigger appropriate approaches to handle it.

INTRODUCTION

The use of social media, such as Facebook or Twitter, has become an increasingly common trend in recent years, and it is most popular amongst teenagers. A major problem concerning teens using social media is that they are often unaware of the dangers involved when using these media.

Cyberbullying is the misuse of technological devices in a manner that causes harm or humiliation to other individuals. Teenagers are more inclined to misuse social media because they are often unaware of the privacy rights associated with the use of that particular website, or the rights of the other users. As a result of this, teens may use social media as new locations to carry out bullying.

We now see a steady rise in cyberbullying incidents because there has been an increase in virtual communication amongst teenagers. Teens may feel more comfortable using social media to bully others because they may feel that they are less likely to get caught. They many times do not think about the ramifications of their actions, and may feel that any harm caused by their actions may never be linked back to them. This false sense of security comes from the lack of adult monitoring of their online activities. Parents may not be home while the teenagers are online, and parents are often too tired to review their child's online history or social media activities. Therefore, the parents are usually unaware that their child has been bullying another child while online. We can concur that this increase will continue as long as the use of the Internet by teenagers continues to increase.

Our best defense against cyberbullying is early detection. However, there are a few challenges that we must consider when developing a system that can detect cyberbullying. One such challenge is that social media varies in how its users interact. A social media may allow its users to follow other users, post content such as messages, articles, images, videos, etc., like or dislike content, or chat to name a few. Therefore, the method used by a bully to carry out the act of cyberbullying may vary from social media to social media, and our system must be able to detect cyberbullying in whichever social media it occurs in. Another challenge in detecting cyberbullying on social media is that some social media allow users to remain anonymous if desired. This can be an obstacle when trying to trace bullying behavior back to a source. Additionally, if a social media allows for user content to be visible to all users instead of selected users, then the cyberbullying may be more difficult to contain, and more challenging to determine how many users are involved. Last, since each social media allows its users to interact and share information differently, we are challenged with creating a standard method for determining which user behaviors may indicate that bullying is occurring. All of these challenges must be taken into consideration in order to develop a system that is efficient in all social media (Hosseinmardi, et al., 2014).

The actions that a bully takes can affect a victim differently, and some actions have a more profound impact on the victim. Therefore, these actions can be divided into different categories based on how severely they may affect a victim. Cyberbullying methods such as insulting/criticizing, ignoring, mocking, and flaming may only result in minor harm or humiliation to a victim. Other methods such as gossip, the spread of unwanted images, impersonations, or videos of the victim being abused may severely affect a victim and should therefore be treated with urgent attention. Understanding the behaviors of a cyberbully will help us design a system that can detect suspicious activity from a user based on common approaches that are often used by cyberbullies (Brush & Helley, 2014).

In this study, we will discuss some common techniques that are often used by a bully to harm or humiliate the victim or victims. Online disputes or arguments between the bully and the victim where some forms of offensive messages are exchanged are known as flaming. These types of disputes can range from minor to severe and may involve inappropriate languages. A victim may experience some forms of harassment where the bully sends hurtful and/or humiliating messages continuously. Rumors about the victim may be sent as messages to outside parties known as denigration. The bully may try his or her best to prevent the victim from participating in online

groups. This cyberbully technique is called exclusion. Personal information or unwanted information, videos, or images about the victim maybe shared with outside parties against the victim's will, and we call this technique outing. Another technique that may be used by the bully is called impersonation in which the bully attempts or successfully hacks into the victim's account to send messages as if they are the victim. Last, the bully may intentionally post or send messages about a particular subject that the victim or others may find offensive. We often see one or more of these techniques being used in cyberbullying cases, and by knowing these techniques we will have some insight into which types of content may be being used by a bully based on the technique that is being used (Willard, 2007).

We will be able to stop cyberbullying with early detection. In this study, we aim to create an effective algorithm that can be used to detect cyberbullying in social media using content mining. Bullies may not use only one social media to victimize another user; he or she may cross many social media to bully the victim. Therefore, we must be able to detect whether or not a user is victimizing someone through one or more accounts, then determine which social media accounts are being used to carry out the victimization. This algorithm will collect information from content shared by the user in all of his or her social media accounts, then extract any information that may indicate that a user is victimizing another user. We'll determine which content to extract based on a big data technology involving phrases or words that might be used by cyberbullies. Any extracted data will give us some insight into whether or not cyberbullying is occurring and provide us appropriate approaches to handle it.

MATERIALS AND METHODS

One of the most well-known algorithms that is currently being used for processing big data (Big Data, n.d.) and generating meaningful datasets is Google's MapReduce algorithm (Dean & Ghemawat, 2004) (Krzyzanowski, 2011) (Zhao & Pjesivac-Grbovic, 2009). MapReduce solves large scale problems by using two operations called map and reduce.

- Map: The mapping function takes in parameter inputs and then divides them into distinct data sets. It then processes these datasets by performing tasks on them.
- Reduce: The outputs of the map function are sent as parameters to the reduce function where the results will be generated.

MapReduce works primarily through the use of computer clusters and/or parallel processing. In a computer cluster, one of the computers within the cluster serves as a manager to the other computers. This manager is responsible for breaking down complex computations into smaller computations, and then distributing the work load amongst the other computers, or worker computers. The computers within this group are usually connected to each other via a local area network (Computer Cluster, n.d.). By using a cluster, we can greatly speed up the processing time. Another advantage of using a computer cluster is that we are able to share the resources of all systems that are connected in the cluster. Parallel processing uses the same concept of breaking complex computations into smaller tasks, and then distributes these tasks amongst different processors within the same system or within the same network (Parallel Processing, n.d.). The manager system distributes the work load for the map function by dividing the input

into a set. The elements within the set are called splits. The number of splits in a set is depended upon how large the data file is. The reduce function distributes the work load by dividing each key space.

MapReduce provides a feature that allows its users to count events in different scenarios. If a user wants to use the counter feature, then he or she would create a counter object that can be incremented periodically in the map function, reduce function, or both. This feature can be useful in solving many different problems. The MapReduce algorithm actually automatically keeps some counter variables in its library for the processing of key/value pairs. It will be useful for our algorithm since we will need to keep track of the number of bullying words that occur in a given file, or the number of webpage files that are being used by a bully.

RESULTS

The proposed solution uses MapReduce function to determine if a web file contains any content that may indicate that it is being used for cyberbullying, it then checks to see if any other social media is being used by the same person for cyberbullying. The solution is explained by the following two cases.

Case 1: Bullying in a single social media

The first case in our solution focuses on the number of bullying word occurrences found within a single social media. In order to give a general understanding of how our algorithm would accomplish this, we will use an example to explain it.

Suppose Bob Smith has a Twitter account, and he has written tweets on ten days in the month of June, 2016. Table 1 shows a chart of June, 2016 and the days that Bob posted tweets are highlighted in yellow.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Table 1. A chart showing the days of June 2016 that Bob posted tweets.

Suppose we would like to check Bob's Twitter page on Wednesday, June 22, 2016 for possible cyberbullying. Let *k* be the minimum allowed cyberbullying words that can be present in a Web file without signaling an alert. Whenever *k* is exceeded, an alert message is sent to the system administrator signaling that cyberbullying may be occurring, and also the user's username, first name, and last name will be included in this message. The algorithm works by making comparisons based on a list of common cyberbullying terminologies. Figure 1 shows how the map function works in this example. Bob's Twitter page is first partitioned into three splits. The number of splits may vary depending on the number of available machines within the cluster.

In this example, suppose the first split contains 3 bullying word occurrences, the second split contains 3 bullying word occurrences, and the last split contains 2 bullying word occurrences. The map phase outputs < key, "1" > for every bullying word found. The "1" indicates that that particular word was discovered from the list of cyberbullying terms and it accounts for 1 occurrence. In the map phase, the manager system segments the input into separate data sets, then the data is sent to available nodes within the cluster. The nodes will then complete the calculation. What we end up with are intermediate < key, value > pairs. The algorithm then sorts and merges the < key, value > pairs that are of same type before the data enters the reduce phase. In this figure, we can see that each < key, value > pair that has the same key is sorted and merged together. We end up with four different < key, value > pairs after the shuffle phase completes. The reduce phase sums up the number of < key, value > pairs that was computed is summed by the manager system. The result is 8 because we ended up with 8 < key, value > pairs total. The algorithm then determines if any cyberbullying is occurring based on the result value. If the result value is greater than *k*, which is the minimum amount of cyberbullying words that are allowed in a given file before it is considered suspicious, then we will signal an alert and store the first name, and last name in a list called names that will be used later to find other social media that may be involved. Also, a count value is incremented from 0 to 1 to indicate that this is the first Web file that we have found from this particular user that may be being used for cyberbullying. The procedures of this case are summarized below:

i. The map function receives the most recently saved version of Bob's Twitter page. This file is sent as a parameter to the map function.
ii. The Web file is then divided into data blocks called splits. The number of splits varies because it is dependent upon how many machines are available within the cluster.
iii. We use a variable, *k*, to represent the minimum amount of cyberbullying terms that are allowed in a single file in order to not be considered suspicious.
iv. The manager system, or master, determines which processors within the cluster are not busy and then assigns them a split.
v. Each processor compares every word within its assigned split against a list of common cyberbullying terms. For every match found, the term and the number 1 is returned as an intermediate <key, value> pair. The key is the term that was found and the value is 1 because 1 match was found.
vi. Each key value goes through a partitioning process in which it is assigned to an available processor for the reduce function.
vii. The <key, value> pairs are sorted into groups that have the same key and value.

viii. The reduce function sums all of the values together for our final result. If this number is greater than *k*, then an alert message is sent and retrieve user information.
ix. The list of suspicious cyberbullying users created.

Case 2: Bullying across several social media

The second case in our solution focuses on identifying other social media that the user may also be using for cyberbullying. The goal is to find data from the first social media Web file that could be used to link the same user to other social media. The way that we can do this is by retrieving the user's first name and last name, then store them in a list that will be used for comparison. This is achieved in the first case. Now we must compare this list against social media user account profiles. We will need to make a comparison against every social media user account profile in order to see if we can find any first and last names that match the names in the linked list.

Suppose Bob has used more than the minimum allowed bullying words on his Twitter page. An alert message has been sent to administrators, and Bob's name has been retrieved. We will now compare Bob's information to other social media. An example of what types of input we may have is shown in table 2.

Facebook	YouTube	Pinterest	Google+
User1: Account Profile	User1: Account Profile	User1: Account Profile	User1: Account Profile
User2: Account Profile	User2: Account Profile	User2: Account Profile	User2: Account Profile
User3: Account Profile	User3: Account Profile	User3: Account Profile	User3: Account Profile
…	…	…	

Table 2. A table showing an example of what the input would look like. Each column represents all the user account profile for a given social media.

In this example, four popular social media account profiles are involved. All social media user account profile Web pages will be sent to the user program as input.

As shown in table 2, the map function will take as input a list of all account profiles from all social media. Suppose that the user account profiles are in alphanumerical order based on usernames. For any account that has the same first and last name, we take the username of that account. Our key will be the name of the social media and the value will be the user's username.

Our output will be a list that consists of the name of the social and the user's usernames. Figure 2 below shows what the map phase would look like.

The input social media Web files will be split into sets files that will be distributed to any available workers. The map phase will check each Web file for any users that have the first name Bob and the last name Smith. For every Web file that satisfies this condition, the name of the social media (key) and the user's username (value) associated with that profile will be taken.

The < key, value > pairs for the Shuffle phase stays the same because each username is unique since some social media may not allow a user to create a username that is already in use. The Reduce phase takes each key value pair and stores the value in a linked list. The output will be a linked list with a pointer called SocialMedia that points to the first node. After all social media have been tested, the list will contain all matches uncovered.

The last procedure of this case takes the list of social media account users and checks whether or not any of them have been used for cyberbullying. Each element in the list SocialMedia is taken, and the user's content page associated with that particular account is tested for cyberbullying words. For example, suppose figure 3 represents our list SocialMedia retrieved from the previous procedure.

Every social media account whose URL is present in the list will have its associated content page entered as input in case 1 to be tested for bullying words. If *k* is violated, then an alert message will be sent for that account to administrators. Since our count variable is no longer 0, we will not test each social media account that has been determined suspicious for first and last names. Nor will we test for other social media accounts that cyberbullying could be occurring on by the same user again. When our count variable is 1, that lets us know that we have already done that for this user. The steps for checking this condition is explained in case 1. The procedures of this case are summarized below:

i. All user account profile Web files will be used as input.
ii. The Web files will be divided into sets that can be distributed amongst workers.
iii. The master system then selects worker systems to assign the splits.
iv. Each worker that is assigned a split will test all Web files within the split against the list of suspicious cyberbullying users that contains the user's first and last name. The file must contain same names for both the first and last name in ordered to be considered a match. If they both match, then we will retrieve the name of the social media as our key and the user's username as the value.
v. Each key value pair goes through a partitioning process in which it is assigned to an available worker for the reduce function.
vi. The reduce function will store all key value pairs into a linked list.
vii. Takes the list of URLs and checks whether or not any of them have been used for cyberbullying

DISCUSSION

The increasing number of cyberbullying incidents that are now occurring could be prevented through the use of big data algorithms such as Google's MapReduce. The algorithm that we have designed can effectively stop cyberbullying incidents while they are still in the early stages. We can greatly decrease the chances of a victim being severely harmed. Some problems that may arise with our algorithm is that it may be difficult to determine who the bully is by Web content if that person does not have a username. Some social media does not require all of its users to have a username. For example, some forums allow users to make comments to articles anonymously. The system would then have to rely on any names that was obtained from that particular Web file. Also, if the system is using a list of common names to determine whether or not a word is actually a name, then the system would only be able to detect common names. If a name is used that is uncommon, then the system will distinguish it as a name. Therefore, we may lose some valuable information about who the cyberbully, victims, and bystanders are. This algorithm could be enhanced in the future rendering it more effective at identifying cyberbullies, victims, and any bystanders.

This algorithm could be improved in the future to render it more effective. Our algorithm can be applied to a cluster of computers or cluster of processors in order to test how well it performs. We could also test its efficiency to see how much time it takes to process small and large scale calculations. The data used within this algorithm can be enhanced for sharper precision. For instance, we would need to have a very large list of cyberbullying terms in order to be able to detect whether or not a Web file is being used for cyberbullying. This list would also need to be updated periodically to ensure that it stays consistent with current cyberbullying techniques. We can improved this algorithm if we add mechanisms that can check for other information that may be present in Web content that could be used to trace the cyberbullying acts back to their perpetrator. For instance, we could add features to the algorithm that will allow it to check for contact information such as email addresses or phone numbers. We could then use this information in an alert message for the administrators leaving it up to them to find the perpetrator or victim, or we could modify the algorithm and have it trace the contact information itself. We can also modify our algorithm to include linked list that contains bullying words from other languages and common names used by other cultures. These are some of the ways that this algorithm could be enhanced in the future.

LITERATURE CITED

Big Data. (n.d.). Retrieved from https://www.techopedia.com/definition/27745/big-data

Brush, S. R., & Helley, H. (2014). *Cyberbullying: Characteristics, Administrators' Responsibilities, and Effective Communication Strategies.* Sanit Louis University. Retrieved March 10, 2016, from http://gradworks.umi.com/36/31/3631288.html

Computer Cluster. (n.d.). Retrieved from Techopedia: https://www.techopedia.com/definition/6581/computer-cluster

Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. *OSDI'04: Sixth Symposium on Operating System Design and Implementation.* San Francisco, CA.

Hosseinmardi, H., Li, S., Yang, Z., Lv, Q., Rafig, R. l., Han, R., & Mishra, S. (2014). A Comparison of Common Users across Instagram and Ask.fm to Better Understand Cyberbullying. *IEEE Fourth International Conference on Big Data and Cloud Computing*, (pp. 355-362). Sydney, Australia.

Krzyzanowski, P. (2011, November). *MapReduce A Framework for Large-scale Parallel Processing.* Retrieved July 28, 2016, from https://www.cs.rutgers.edu/~pxk/417/notes/content/mapreduce.html

Parallel Processing. (n.d.). Retrieved from Techopedia: https://www.techopedia.com/definition/4598/parallel-processing

Willard, N. (2007, April). *Educator's Guide to Cyberbullying and Cyberthreats.* Retrieved April 6, 2016, from http://www.cyberbully.org/cyberbully/docs/cbcteducator.pdf

Zhao, J., & Pjesivac-Grbovic, J. (2009). MapReduce: The Programming Model and Practice. *SIGMETRICS'09 Tutorial.*

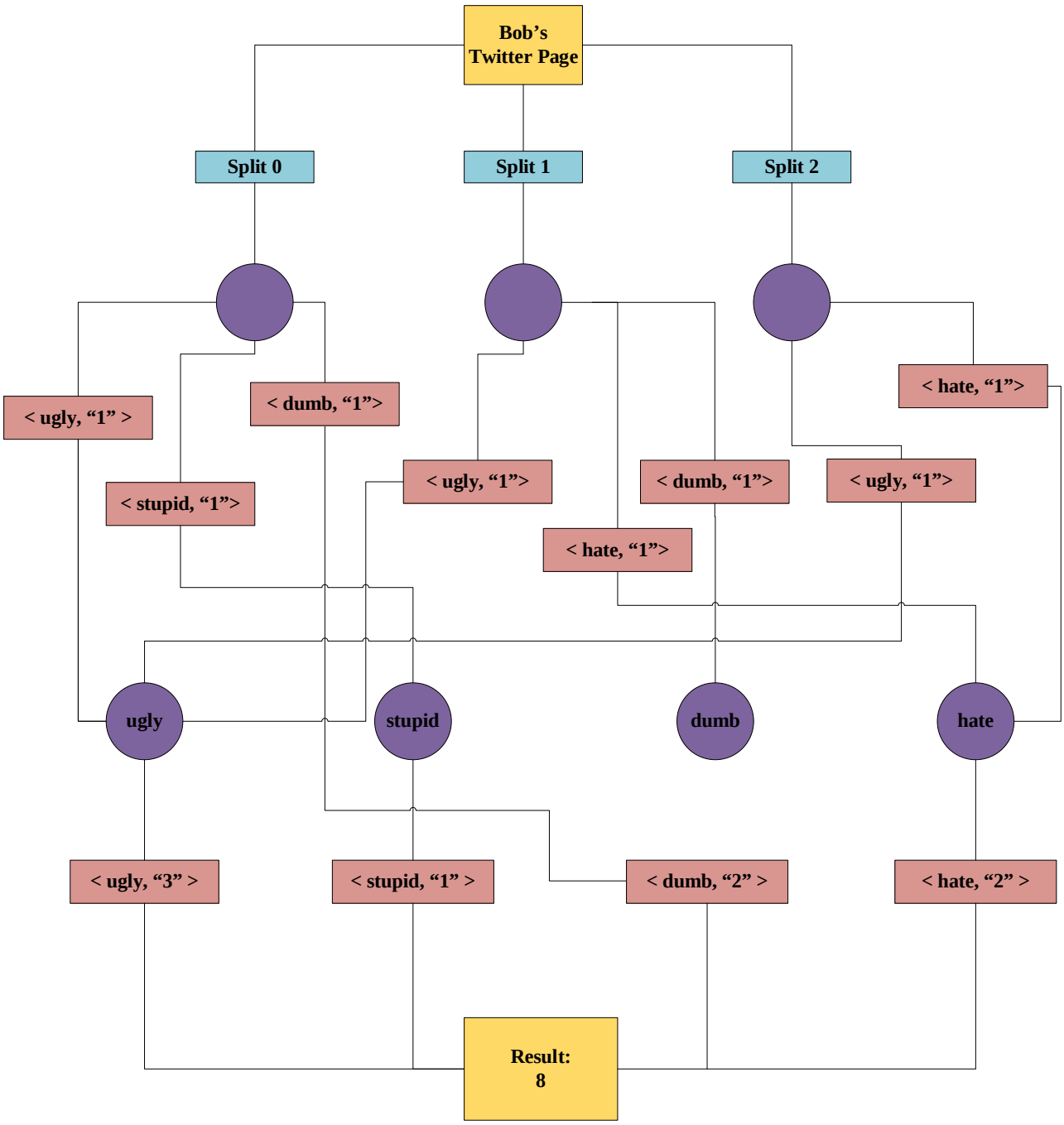

Figure 1. A graph of the map and reduce functions for case 1.

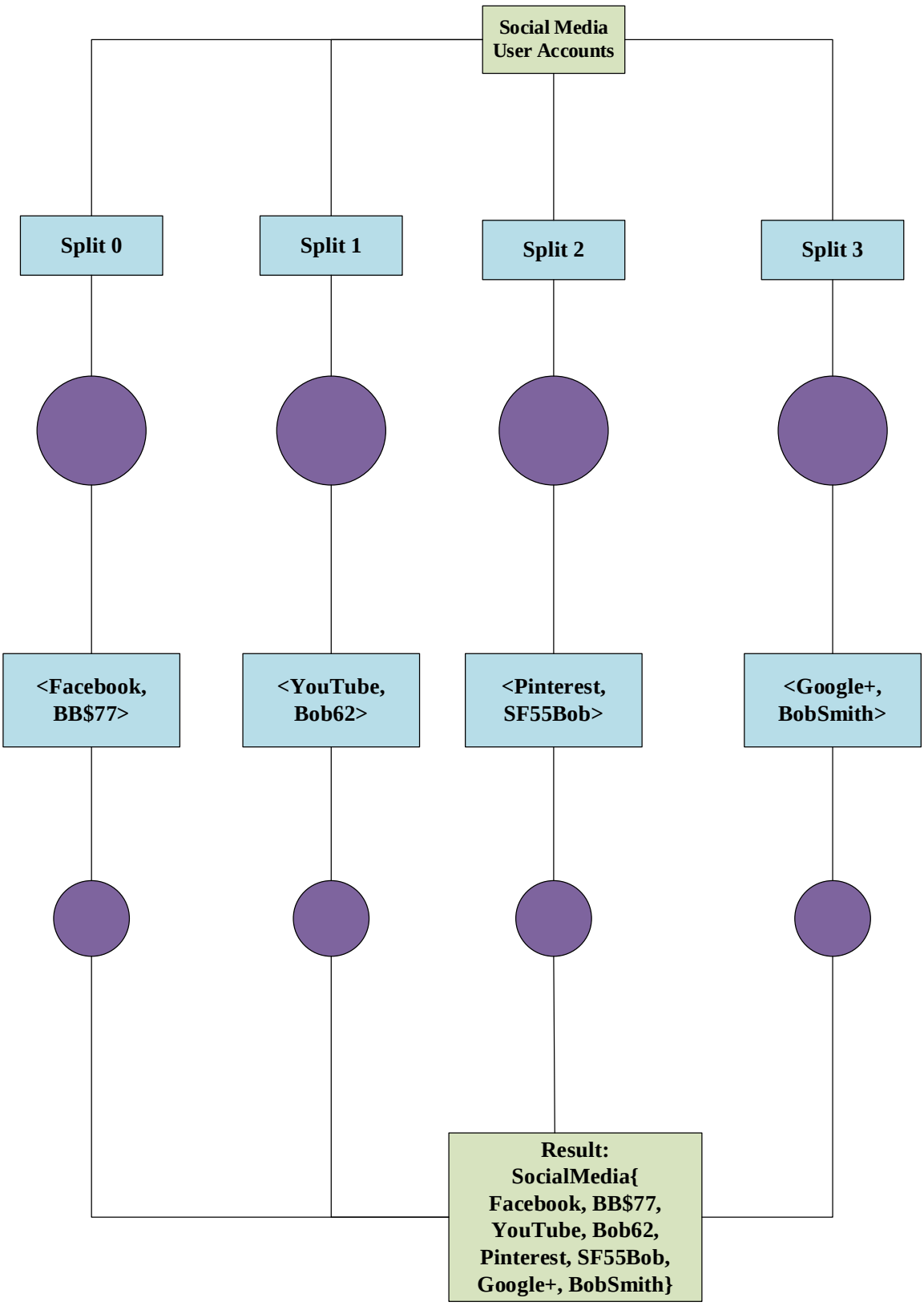

Figure 2. A graph of the map and reduce functions for computation 2.

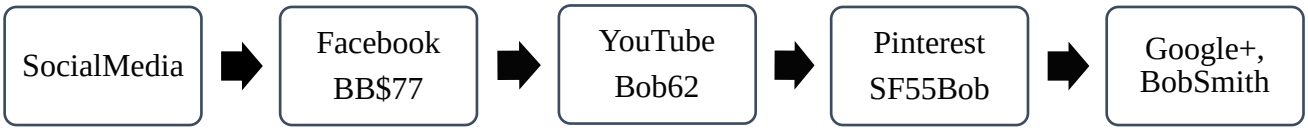

Figure 3. A diagram of the linked list Social Media from case 2.