Social Spreadsheet

Juan Jara, Fabio Casati, and Maurizio Marchese

University of Trento, Dept. of Information Engineering and Computer Science Via
Sommarive 5, 38123 Povo (TN), Italy
{juan.jara, casati}@disi.unitn.it, maurizio.marchese@unitn.it

Abstract. Social media data is growing exponentially, to the point where it is already hard to analyze. Consequently, there is a need to increase the number of people analyzing this data, to make sense out of it and, if possible, react to it. However, accessing this data is not simple because it is behind a knowledge barrier, which can only be overcome either with learning or with money. To considerably lower this barrier, we implemented the Social Spreadsheet, which is a spreadsheet template that we extended with functions that make simple the retrieval of social media data. Moreover, the collected data is ready to be analyzed by end-users, who can use formulas, custom functions, charts, and other commonly known spreadsheet features to create visualizations similar to the ones offered by commercial applications. To validate our work, we demonstrate how end-users can easily implement the same dashboards as the ones offered by popular social media analysis tools.

Keywords: Spreadsheet-based Applications, Social Media Analysis, End-user Programming

1 Introduction

The adoption of smartphones exploded and, with it, the use of social networks, which in turn caused an exponential growth of the already available social media data. Social media data is the data that is obtained from the different social networks like Facebook, Twitter, YouTube and others. The types of social media data that are most used are the ones that indicate demographic information, location, interactions and user preferences.[1]

People from different fields (marketing, health, sociology, etc.) are looking forward to access this data, to analyze it and use it for different purposes like performing prediction analysis, improving advertising campaigns, getting feedback for a service or product, obtaining communities insights and so on.

Currently, the people interested in accessing to social media data have to choose between two options to get it. The first option is to get the data through one of the multiple available web applications like Simply Measured [1], Social Bakers [2], Sprout Social [3] or other social media analytics web applications.

[1] http://www.emarketer.com/Article/Marketers-Use-Social-Media-Data-Drive-Campaigns/1009682

The problem is that these applications usually charge a fee for the data or put some constraints to the request or to the provided data. The second option is to develop an application that will get directly the data from the social network. The problem with this option is that is available only to the people with programming knowledge and, moreover, most social networks (like Facebook) do not allow data extraction for personal use.

There are a few applications for social network analysis that get data for free from social networks, like NodeXL [4]; unfortunately, this data is specifically for social network analysis and thus, is of little use for doing other types of analysis.

In this context, we implemented the Social Spreadsheet, which is our approach to provide a user-friendly environment for working with social media data.

In order to meet our objectives, we need to overcome some challenges:

- *The representation of social network entities in the spreadsheet tabular layout:* A **social networks entity** is a complex data, it can have single-valued attributes (like name, gender or birthday), multi-valued attributes (like spoken languages or interests) and it can be related to other entities (e.g., a user has friends, photos, etc.). The challenge here is defining a mapping for these complex entities into the simple typed cells (string, number, date) of the spreadsheet and its tabular layout.
- *Abstract the social network API (Application Programming Interface) model into the spreadsheet function model:* we need to use spreadsheet functions to connect to social networks APIs and get data from them. Some of the characteristics of spreadsheet functions (like cell referencing or automatic refresh) can affect negatively the user experience (like unnecessary or unwanted calls to the social networks APIs). The challenge here is in deciding which characteristics we want to keep, change or avoid to provide a user-friendly experience without deviating too much from the spreadsheet paradigm.

In our proposal, we define custom functions that abstract from the users the calls to the social networks APIs. These functions, which are common to the spreadsheet paradigm, provide to users a familiar method to access social media data. The combination of these functions and the spreadsheet features creates an environment where users can create visualizations similar to the ones found in the market. More specifically, with the Social Spreadsheet we provide:

- A **conceptual model** for representing the entities for which we can get data from the social networks,
- A **set of functions** that abstract the social network API model and that offer to spreadsheet users a more familiar method for getting social media data, and
- A **process** for providing extra-features that support the reuse of user-designed visualizations and creation of time series reports and charts.

We provide all the above in the well-known spreadsheet environment. Spreadsheets are intuitive, easy to use and are specifically designed for the manipulation and visualization of data.

The remainder of this paper is organized as follows. In the next section we explain our motivation for doing this work. In Section 3, we explain the state of the art on obtaining data from social networks and presenting it on spreadsheets. In Section 4, we present our proposal for a Social Spreadsheet. In Section 5, we describe how we implemented our proposal. In Section 6, we introduce some related work and compare them to our proposal. In Section 7, we validate our approach by implementing data visualizations similar to the ones offered by commercial applications for social media analysis. In Section 8 we discuss future works.

2 Background and Design Principles

People working in health, sociology, marketing and other fields need to constantly analyze data related to their respective target groups and communities [5–7]. The source of this data is usually the result of census, surveys, questionnaires or other similar sources which gather the data manually. Lately, with the boom of social networks, the people who analyze social data noticed that the data gathered by social networks in months was far richer than the data gathered by their usual sources in years.

At this point, it was evident that the interesting data was in the hand of businesses, and therefore, the people interested in working with it had to use the tools provided by these businesses. Sometimes, this situation resulted in people using a tool for doing one type of work when the tool was designed for doing another type, e.g., using a social network analysis tool for doing social media analysis.

Among the most popular tools for the manipulation and analysis of data are the spreadsheet applications. In fact, the characteristics of the spreadsheet paradigm make the spreadsheet an environment where is easy and intuitive for users to view and interact with data [8–11]. Among the spreadsheet characteristics defined in [8], the most important are:

- The **tabular layout** that makes easy to visualize and manipulate collections of entities,
- The **operators** for creating or modifying cell contents, which can be applied to a single cell or to a range of cells (such as an entire row), and
- The **dependencies between cells** that manage the propagation of changes between referenced cells, i.e., when a cell is modified, the content of its dependent cells will be updated automatically.

The work in [12] estimates that by 2012 there will be approximately 55 million spreadsheet users, and this is only in the United States. This does not mean that all these people will benefit from our proposal, but for the ones that will work doing social media analysis, our proposal will be of great help and will lower the entrance barrier to the field of social media analysis.

Spreadsheets are one of the most successful end-user programming applications. For example, a user creates a spreadsheet and uses formulas to solve some calculations. Later, the user can reuse the spreadsheet for new calculations just by changing the values of the input cells, that is, the user is able to abstract the algorithm from the data.

On the basis of the above considerations, in our work we focused our attention on the spreadsheet paradigm. Our goal is to give users the ability to create spreadsheet reports, charts and visualizations similar to the ones available in the commercial tools by facilitating to these users the access to the data that they need (social media data) in a spreadsheet environment (considered the most suitable environment for doing data manipulation and analysis).

3 State of the Art

In this section we explain the works and proposals that we found related to the retrieval of social media data from social networks and the presentation of this data on spreadsheets. We start explaining the methods for representing social media data in the tabular layout of the spreadsheets and then, the different approaches for using custom functions to call social networks APIs.

3.1 Represent Social Media Data in Spreadsheets

To copy data from social networks to spreadsheets, we need to define a mapping that indicates to the application how to put the obtained data into the spreadsheet cells. We found two approaches for defining these mappings in the applications and proposals that we studied:

- The first approach defines a straightforward mapping of the entities to the spreadsheet cells, i.e., each entity attribute is mapped to one cell. If the entity has an attribute that represents a collection, the attribute is mapped as a new entity and printed separately (usually in a different sheet). Most of the tools and proposals that we found use this approach, e.g., [4, 1].
- The second approach defines a mapping where a single cell contains the representation of a whole entity (with its instances). The data of the entity can be accessed using a formula language that allows a cell to reference a specific attribute of a specific instance of the entity. A detailed explanation of this approach can be found in [11, 13].

3.2 Call APIs with Spreadsheet Custom Functions

In the spreadsheet paradigm, a formula or function can take as input the reference to a cell or to a range of cells (this is besides the normal parameters that are usually passed to the function). The result or output of the function has a similar behavior; the result can be a single value that will be printed in the same cell where the function is defined or, the result can be a set of values that will be printed in the cell where the function is defined plus the cells to its bottom and its right (depending on the range of the result). In some of the analyzed works, we found interesting proposals for custom functions that combine the previous input/output options with the interconnection to other applications APIs.

In [10], the authors define (single cell input and single cell output) custom functions that use the cell value to call applications APIs. The authors then propose to define data flows between services by making custom functions reference cells that contains other custom functions and thus, allowing users to easily construct service mashups. If the input of the first function in the data flow is modified, the rest of the values in the data flow will be updated automatically due to the dependency between cells, a well-known characteristic of the spreadsheet paradigm.

The work in [11] allows users to associate a function with a web data service (the data service represents a single entity). Each time the function is used it will get from its related source a collection of entities (a function with no cell input but a multi cell output). The entity attributes, the number of items in the collection and how the data will be visualized are configured by the user. What is interesting from this approach is the implementation of a framework that supports the update of the data on the spreadsheet whenever the data in the source changes. The framework also supports the update in the other direction, i.e., when the data in the spreadsheet is modified by the user, the corresponding data in the source is also updated, when possible.

The functions in both cases take into account the active cell when they print their results and thus, they follow the standard behavior of functions of the spreadsheet paradigm. The work in [4] implements functions that do not follow this standard behavior. These functions get data from applications APIs and print their results in cells predefined by the functions algorithms without taking into account the active cell position.

4 Social Spreadsheet

Our proposal consists of extending spreadsheet-based applications with custom functions for accessing social media data. We think that the combination of these functions with the characteristics of the spreadsheet paradigm creates a user-friendly environment for the design and creation of social media data visualizations.

We start with the presentation of a conceptual model for representing the social media data that we get from social networks. Then, we present the functions that connect the entities from our model with the social network APIs. Finally, we introduce two special templates; one for recording the execution of functions in a script that can be executed later to produce the same layout using different data sources; the other for computing periodically metrics that will allow users to easily create time series charts and reports.

4.1 Social Media Conceptual Model

In this section we focus on the creation of visualizations for social media analysis, and thus, we limit the data that we get from social networks to the data that is most useful for this type of analysis. We designed our conceptual model based on the Facebook API model[2] because at the moment Facebook is the most used social network.

We show our social media conceptual model in Fig. 1. The entities that we abstracted are the following:

- **User:** this entity represents the current user (the user that is using the tool). Some of the attributes that we get for the current user (if available) are its name, gender, current location, birthday, relationship status, etc. The user can have friends, albums, photos and posts.
- **Friends:** this entity represents a social network user that has a friend connection with the current user. The attributes that we get for friends are the same attributes that we get for the current user. The friends can have albums, photos and posts.
- **Pages:** this entity represents a business or brand profile in a social network. Some of the attributes that we get for pages (if available) are their name, page link, page category, and business website. Moreover, we get two attributes that are very popular in the social media analysis field, the likes count and the people talking about this count.[3] The page can have albums, photos and posts.
- **Albums:** this entity represents a photo album defined by a user or page. Some of the attributes that we get for albums (if available) are their name, photo count, description, link, cover photo, etc. An album can contain photos.
- **Photos:** this entity represents a photo uploaded to the social network by a user or page. Some of the attributes that we get for photos (if available) are their name, link, time of upload, etc.
- **Posts:** this entity represents a message post created by a user or page. Some of the attributes that we get for posts (if available) are their name, type, message, link, etc.

Figure 1 shows the entities that we model from social networks and how they relate to each other. We consider these relations when we define the spreadsheet functions that we present in the next section. We do not do a complex mapping of the entity with the spreadsheet cells like in [11], we just map each entity attribute to one cell.

The conceptual model was designed based on the Facebook entities but it could be used to some extent to get the entities of other social networks. For example, social networks like twitter and Google+ have the user entities which are similar to our user entity and the attributes of these entities can be mapped to the attributes of our model. Also, our post entity is similar to the activities entity of Google+ or the Tweets entity of Twitter. For the friends entity we need to extend the model because in Twitter and Google+ this relationship is

[2] http://developers.facebook.com/docs/reference/api/

[3] For more details about these attributes check http://developers.facebook.com/docs/reference/api/page/

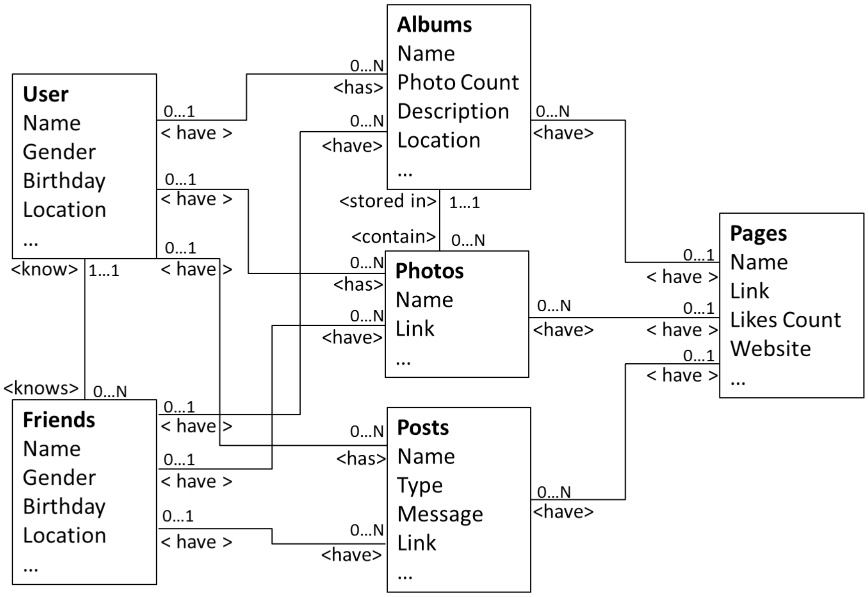


Fig. 1. The social media conceptual model

asymmetrical (if user A is a friend of user B it does not mean that user B is a friend of user A).

4.2 Spreadsheet Social Functions

We defined two types of custom functions for getting data from social networks into spreadsheets. The data functions, which are functions used to get the instances of an entity from the social network, and the metric functions, which are functions that computes a metric for an instance of an entity.

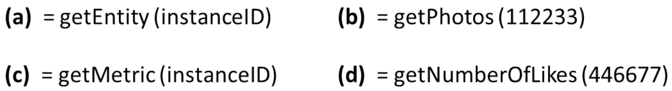
Figure 2(a) shows a generic form of a data function. The characteristics of this type of functions are the following:

- **Name:** the name of the function indicates the type of the retrieved instances, e.g., friends, photos, albums, etc. The entities available for retrieval are the ones that are present in the conceptual model.
- **Input:** this type of functions takes one input parameter that represents the *id* of the owner of the retrieved instances, e.g., in Fig. 2(b) the function will get all the photos for the user which *id* is 112233. If the *id* is null or empty the function will get the instances that belong to the current user.
- **Output:** this type of functions produces a multi-valued result. The range of the result depends on the number of attributes (one column per attribute) and the number of instances (one row per instance) that is retrieved.

- **Cell dependency:** cell dependency is not activated for this type of functions, i.e., the API to get the instances will be called only one time, if the input parameter changes the API will not be called again and the results will still correspond to the old input.

Figure 2(c) shows a generic form of a metric function. The characteristics of this type of functions are the following:

- **Name:** the name of the function indicates the metric that will be computed, e.g., number of likes, number of comments, number of female friends, etc. The available metrics were implemented according to the most popular metrics found in several social analysis tools.
- **Input:** this type of functions takes one input parameter that represents the *id* of the instance for which the metric will be computed, e.g., in Fig. 2(d) the function will get number of likes for the photo which *id* is 446677.
- **Output:** this type of functions produces a single-valued result.
- **Cell dependency:** cell dependency is activated for this type of functions, i.e., the metric will be computed each time the input value changes.



(a) = getEntity (instanceID)

(b) = getPhotos (112233)

(c) = getMetric (instanceID)

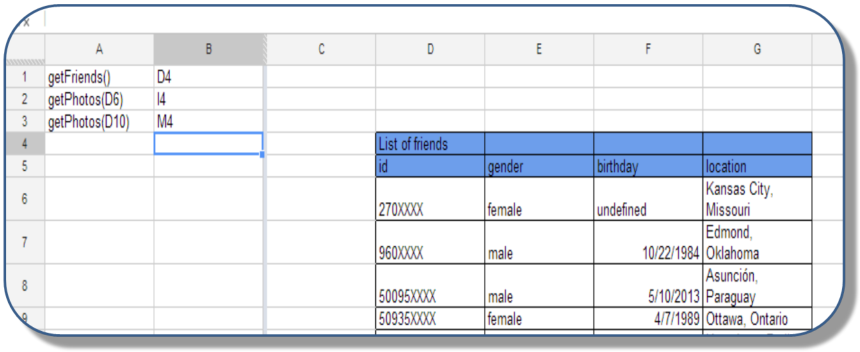
(d) = getNumberOfLikes (446677)

Fig. 2. (a) Generic data function. (b) Data function that gets the list of photos for the friend with id=112233. (c) Generic metric function. (d) Metric function that counts the number of likes for the photo with id=446677.

4.3 Special Purpose Templates

As examples on how to support the users in the manipulation and visualization of social media data we designed two sheet templates; one to support the automatic creation and reuse of user visualizations and, the other to support the creation of time series reports and charts.

The *template that supports the creation and reuse of user visualizations* uses the first and second column of sheet. The first column is used to record all the user actions (which include data manipulation and execution of functions) and the second column is used to store the position where the action occurred. Figure 3 shows an example of this template. The purpose of this template is to create a script that describes the process of creation of a visualization layout. The script can be used later to generate the same layout but using a different data source, this is done by changing either the executed function or the function parameter. This feature can be used to compensate the lack of cell dependency of the data functions.



	A	B	C	D	E	F	G
1	getFriends()	D4					
2	getPhotos(D6)	I4					
3	getPhotos(D10)	M4					
4				List of friends			
5				id	gender	birthday	location
6				270XXXX	female	undefined	Kansas City, Missouri
7				960XXXX	male	10/22/1984	Edmond, Oklahoma
8				50095XXXX	male	5/10/2013	Asunción, Paraguay
9				50935XXXX	female	4/7/1989	Ottawa, Ontario

Fig. 3. Template for recording and executing action scripts

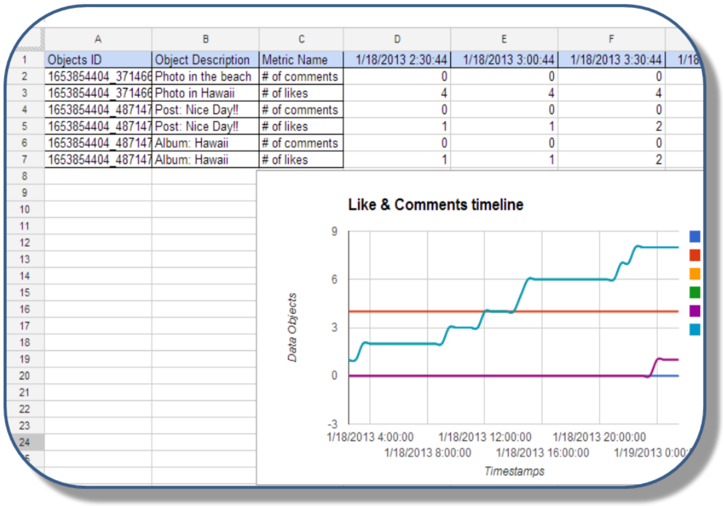
The *template that supports the creation of time series reports and charts* uses the first row and, the first to third columns. The first column is used to store the ids of the entity instances for which a metric will be computed. The second column is used to store the description of the entity instances whose ids are in the first column, this will save time to the user that will not need to search all over the spreadsheet to which instances correspond the ids in the first column. The third column is used to store the metrics that will be computed for the entity instances of their corresponding rows. This template has an associated function that is executed periodically (the interval is defined by the user). The periodic function, on each execution, gets the first not used column and prints the current timestamp on the first row of that column. Then, for each row, executes the metric function specified in the third column using as input parameter the id specified in the first column. Finally, the result of the metric computation is printed on the column where the last timestamp was stamped. Figure 4 shows an example of this template. After some executions of the periodic function the template will have the necessary data for the creation of time series reports and charts.

5 Implementation

We implemented the Social Spreadsheet by extending a Google spreadsheet with the functions described in Sec. 4.2 using the Google App Script scripting language. The spreadsheet is public and can be used by anyone with a Google account.

We published the Social Spreadsheet in the Google template gallery[4]. The template can be found by typing Facebook Analytics Template in the search template box. To use the template, users need to get an access token from our

[4] https://drive.google.com/templates



	A	B	C	D	E	F	
1	Objects ID	Object Description	Metric Name	1/18/2013 2:30:44	1/18/2013 3:00:44	1/18/2013 3:30:44	1/18
2	1653854404_371466	Photo in the beach	# of comments	0	0	0	
3	1653854404_371466	Photo in Hawaii	# of likes	4	4	4	
4	1653854404_487147	Post: Nice Day!!	# of comments	0	0	0	
5	1653854404_487147	Post: Nice Day!!	# of likes	1	1	2	
6	1653854404_487147	Album: Hawaii	# of comments	0	0	0	
7	1653854404_487147	Album: Hawaii	# of likes	1	1	2	

Fig. 4. Template for computing metrics periodically

application website[5]. In the next section we explain how to get the access token and why it is needed.

5.1 Authentication with Social Networks

In order to interact with social networks, applications need to authenticate themselves with the social networks. If the authentication process is successful, the social network gives an access token to the authenticated application. The application can now interact with the social network passing with each request the given access token.

In our implementation, the authentication process is implemented outside of the spreadsheet. We did not include the authentication process in the spreadsheet due to a limitation of the scripting language at the moment of the implementation of the Social Spreadsheet. We continuously monitor the language updates to check if this limitation has been removed.

To get the access token to make the spreadsheet functional, users need to go to the application website and do the authentication process there, at the end of the process they will get the access token for the spreadsheet. The token expires after two months, after that it can be refreshed just by visiting the application website.

[5] https://comealong.me/fb/sp/socialspreadsheet/

5.2 Google Spreadsheet vs. Excel

Why we choose Google Spreadsheet instead of Excel for the implementation of the Social Spreadsheet? We actually started to implement the Social Spreadsheet for Excel because we consider Excel as a more mature and more complete spreadsheet tool than Google Spreadsheet. However, we found that some critical functions that were available for the Windows version of Excel were missing for the Mac version of Excel. For this reason we changed the implementation tool to Google Spreadsheet.

Implementing the Social Spreadsheet for Google Spreadsheet has its advantages with respect to Excel. The main advantages are that it is free for anyone with a Google account and that it has more portability because Google Spreadsheet is a web application and is available to anyone with an internet connection and a compatible web browser.

Either way, for users that feel more comfortable working with Excel, they can use the Social Spreadsheet to get the all social media data that they need and export it to Excel from the Google Spreadsheet. Then, they can continue working with the data in Excel but without the added support of the Social Spreadsheet functions.

6 Related Work

Social media data is continuously growing with the constant flow of new posts, comments, tweets and likes; all these data is drawing the attention of more and more marketers that want to analyze it for getting consumer insights among other information. This growing interest in social media data did not go unnoticed by the software market, which increased its offer on applications for content analysis, dashboards and reports based on the analysis of social media data.

There is an extensive list of web applications for social media analysis. We selected and investigated some of the applications that are considered the most interesting according to a few blogs and articles that we found on the web.[6][7][8] From the applications that we investigated, the following are the ones that we consider most relevant: Simply Measured [1], Social Bakers [2], Inside Network [14], Sprout Social [3] and Page Lever [15]. All of them offer several reports, charts and dashboards. Table 1 shows the classification of these tools according to the following features:

- **Data provision capability:** indicates to what extent they provide social media data on spreadsheets. The possible values are:
  - *None*, which means that they do not provide data at all;

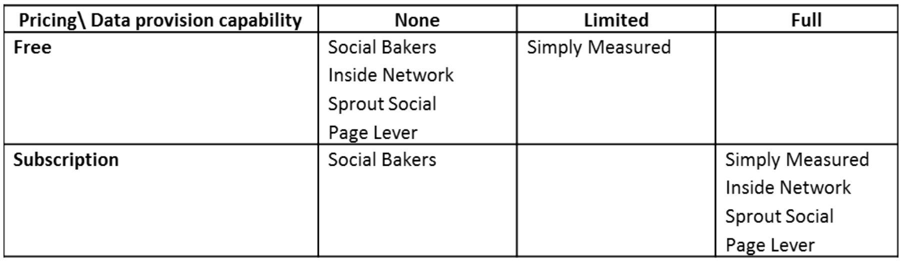
[6] http://www.convinceandconvert.com/social-media-tools/the-39-social-media-tools-ill-use-today/

[7] http://thenextweb.com/socialmedia/2011/09/02/the-best-new-social-media-analytics-tools-of-the-year-so-far/

[8] http://thenextweb.com/socialmedia/2012/03/18/50-mostly-free-social-media-tools-you-cant-live-without-in-2012/

  - *Limited*, which means that they provide data but limited with constraints to the request or to the provided data and;
  - *Full*, which means that they provide all the requested data.
- **Pricing:** indicates under what payment model the data is being provided. The possible values are:
  - *Free*, which means that no money is paid for getting the data and;
  - *Subscription*, which means that a periodic fee is paid for getting the data.

Table 1. Classification of social media analysis tools according to their data provision capability and payment model.



Pricing\ Data provision capability	None	Limited	Full
Free	Social Bakers Inside Network Sprout Social Page Lever	Simply Measured	
Subscription	Social Bakers		Simply Measured Inside Network Sprout Social Page Lever

Currently, the Social Spreadsheet does not provide all the complex charts and reports as the previous classified tools. Instead, the Social Spreadsheet provides the needed data in a suitable and user-friendly environment for the creation of similar visualizations and for free.

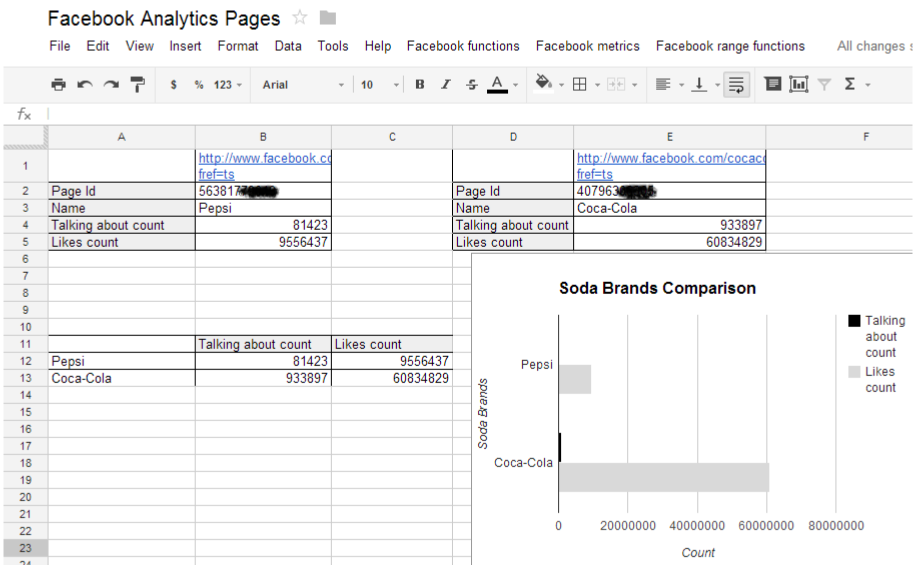
We found other related works that get data from the web to spreadsheets using spreadsheet functions but the scope of these works are different than ours:

- NodeXL [4], which is an Excel template that gets data from social networks but with the only scope of doing social network analysis. Moreover, the entities that can be obtained are fixed; users can only choose the entity attributes they want to get.
- In [10], the authors propose the creation of service mashups by concatenating spreadsheet functions using the cell dependency characteristic of spreadsheets. The defined functions get data from public APIs that do not require authentication and, take a single-valued input and produce a single-valued output.
- Finally, in [11], the authors also propose the creation of mashups using spreadsheet functions. The authors propose spreadsheet functions that get data from public APIs that do not require authentication, take none or a single-valued input and produce a multi-valued output and, propagates data changes between the data source and the spreadsheet (the propagation can be bidirectional if it is allowed by the source).

7 Validation

In this section we validate our proposal by implementing one data visualization that is similar to the ones offered by commercial applications (like the ones mentioned in the previous section).

In this example, we show the process for building a bar chart visualization for comparing two popular soda brands using two metrics, the *likes count* and the *people talking about this count*. Figure 5 shows all the data necessary to build the visualization and the resulting chart. Each step of the explanation will use



	A	B	C	D	E
1		http://www.facebook.cc fref=ts			http://www.facebook.com/cocacc fref=ts
2	Page Id	563817[redacted]		Page Id	407963[redacted]
3	Name	Pepsi		Name	Coca-Cola
4	Talking about count	81423		Talking about count	933897
5	Likes count	9556437		Likes count	60834829
11		Talking about count	Likes count		
12	Pepsi	81423	9556437		
13	Coca-Cola	933897	60834829		

Fig. 5. Bar chart for comparing two soda brands using the *likes count* and the *people talking about this count* metrics.

as reference the data shown in Fig. 5:

1. We copy the links of the Facebook pages of the desired brands in the first row.
2. We get the data for each page by positioning the active cell on the cell where the page link is and then we select the menu item Get Page under the Facebook functions menu. The page data will be copied under the cells where the page link is (as shown by Fig. 5).
3. We put the data in a format that is accepted by the bar chart. We need to put all the brand names as row headers and the metric names as column headers (columns A, B, C and rows 11- 13 of Fig. 5).
4. We select the range of the table created in step 3 and select the option to add a bar chart. The result should be similar to the observed in Fig. 5.

The above 4 steps illustrate how simple is to create one of the most popular visualizations. More examples can be found in our application website[9].

8 Conclusion and Future Work

With our proposal we demonstrated how end-users can create visualizations similar to the ones offered by commercial tools. This also served to demonstrate the great potential of spreadsheet applications as end-user programming tools.

Spreadsheet applications can be easily extended by the addition of custom functions. However, if the targets of the extended application are end-users, the functionality added should be minimal and, if possible, comply with the characteristics of the spreadsheet paradigm. This is to maintain the simplicity and familiarity that end-users are accustomed to find when working with spreadsheets.

The proposed conceptual model is based on the Facebook data model. As a part of our future research we need to study how to abstract or extend more the conceptual model to comply with the data models of other popular social networks like Twitter or Google+. Furthermore, we want to conduct a usability study to evaluate how actual end-users perceive our proposal and to receive feedback of features or functionality that we could have omitted.

Acknowledgments. This work was supported by funds from the BPM4People project (`http://www.bpm4people.org`) of the EU FP7 Capacities program.

References

1. Social Media Analytics, From Data to Deliverables, `http://simplymeasured.com/`
2. Socialbakers: Social Media Marketing, Statistics & Monitoring Tools, `http://www.socialbakers.com/`
3. Sprout Social: Social Media Management, Social CRM for Business, `http://sproutsocial.com/`
4. NodeXL: Network Overview, Discovery and Exploration for Excel, `http://nodexl.codeplex.com/`
5. Mackay, H.: Information and the Transformation of Sociology: Interactivity and Social Media Monitoring. tripleC-Cognition, Communication, Co-operation **11**(1) (2012) 117–126
6. Hansen, D.L., Rotman, D., Bonsignore, E., Milić-frayling, N., Rodrigues, E.M., Smith, M., Shneiderman, B., Capone, T.: Do You Know the Way to SNA ?: A Process Model for Analyzing and Visualizing Social Media Data. Group **56**(3) (2009) 1–10
7. Thackeray, R., Neiger, B.L., Hanson, C.L., McKenzie, J.F.: Enhancing promotional strategies within social marketing programs: use of Web 2.0 social media. Health Promotion Practice **9**(4) (2008) 338–343
8. Chi, E.H., Riedl, J., Barry, P., Konstan, J.A.: Principles for information visualization spreadsheets (1998)

[9] https://comealong.me/fb/sp/socialspreadsheet/

9. Ballinger, D., Biddle, R., Noble, J.: Spreadsheet visualisation to improve end-user understanding. In: Proceedings of the AsiaPacific symposium on Information visualisationVolume 24, Australian Computer Society, Inc. (2003) 99–109
10. Hoang, D.D., Paik, H.y., Benatallah, B.: An analysis of spreadsheet-based services mashup. In: Proceedings of the Twenty-First Australasian Conference on Database Technologies - Volume 104. ADC '10, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2010) 141–150
11. Kongdenfha, W., Benatallah, B., Vayssière, J., Saint-Paul, R., Casati, F.: Rapid development of spreadsheet-based web mashups. Proceedings of the 18th international conference on World wide web WWW 09 (2009) 851–860
12. Scaffidi, C., Shaw, M., Myers, B.: The 55M End-User Programmers Estimate Revisited. Technical Report February, Carnegie Mellon University (2005)
13. Saint-Paul, R., Benatallah, B., Vayssière, J.: Data services in your spreadsheet! In: Proceedings of the 11th international conference on Extending database technology: Advances in database technology, ACM (2008) 690–694
14. Research and Analysis of the Facebook & Mobile App Ecosystems, `http://research.insidenetwork.com`
15. PageLever: Analytics & Social Marketing Tools for Facebook Pages, `http://pagelever.com/`