SocioScope: an Integrated Framework for Understanding Society from Social Data

Hoang Long Nguyen[1], Minsung Hong[1], Na-Yeong Cho[1], Junbeom Kim[1], David Camacho[2], and Jason J. Jung[1⋆]

[1] Department of Computer Engineering, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul, Korea
{longnh238,minsung.holdtime,joenayoung2,dreaminheart,j2jung}@gmail.com
[2] Department of Computer Science, Universidad Autonoma de Madrid, Madrid, Spain
david.camacho@uam.es

Abstract. We recognize the importance of data, especially in the period of Internet of Things. However, effectively collecting social data from different sources and analyzing relationships between them to understand our society is actually a big challenge. SocioScope is built to solve this problem. Besides, we recognize that many researchers are spending time for conducting same work (i.e., collecting and pre-processing data). Therefore, we aim to provide SocioScope as a framework for reducing their effort time. Outputs of our system are not only used for understand about social data but also possible to use as inputs for other work to create useful applications.

Keywords: SocioScope framework, Social data, Social information, Society understanding

1 Introduction

We are living in era of data due to the growth of Internet of Things (IoT). In IoT system, devices (e.g., wireless sensor networks, GPS, control systems) are connected and data is created through every events. By the year 2013, IoT had been integrated into different systems by using multiple technologies. Therefore, data increased quickly in every sides including volume, velocity, variety. Due to a statistics in [2], data reached 4.4 zettabytes in 2013. This brings huge benefits for society.

Data is the facts about our world. There are two types of data which are tacit and explicit data [6]. Tacit data is achieved through experience and is embedded in the human mind. On the opposite, explicit data consists of printed and electronic materials. In this work, we focus on explicit data. Besides, we also concentrate on social data instead of individual data (e.g., all data which is collected from President Barack Obama). Social data refers to the set of data

⋆ Corresponding author.

which is created by different individuals from social media, mass media, and sensor data. However, social data is still raw and discrete. It exists without any context and analysis. It can not be useful itself unless it is processed to obtain the information.

Therefore, we need a system for integrating and analyzing social data to understand its relationships and connections. Then, we get social information which is structured representation of social data. By obtaining social information, we can answer questions which are related to "who, what, where, and when". For example, social data is related to the number of vehicles on street. From analyzing data to get information, we can understand about our society with questions such as: who stay on this street? what kind of vehicles that people are using? where is the available street? and when is the period time of traffic jam? The scope of society is also very dynamics and it depends on social data that we get. Society can be a university, a company, or a community.

Our contribution in this paper is as follows. We build SocioScope for the target of collecting social data and creating social information. From that, we can have our insight into society around us. Besides, we recognize that researchers must spend a lot of time for same work (i.e., collecting and pre-processing data). By providing SocioScope framework, we want to reduce their effort time. The outputs of SocioScope can be received as inputs for other work to create social knowledge, or even social wisdom. Coming back to the above example, we can produce an application to guide for avoiding traffic jam using social information. This is what we call social knowledge [1].

In this section, we give the overview of our motivation. Section 2 will describe our system in detail. Further, the performance of SocioScope will be presented in Section 3. Finally, we conclude and discuss future work in Section 4.

2 SocioScope Framework

2.1 Overview

In this section, we give detail information for well understanding the overview of our system. SocioScope is implemented by using Java programming language and is independently divided into web service and background service (i.e., windows service or linux daemon) as shown in Fig. 1.

The web service side is implemented by using Apache Tomcat servlet container and Java Server Pages technology. We choose Model View Controller (MVC) as our programming pattern for well separating logical handling of data and presentation of the data. Further, we design important functions in background service which is run by command prompt. The advantages of background service are higher system performance and better security. Moreover, it reduces wrong user's behavior. These two systems communicate with each other by using Apache Thrift which is a framework for cross-language services development. Apache Thrift effectively works with diversity programming language (e.g., C++, Java, Python, and PHP). Further, we can take advantages of Apache Thrift when we want to integrate other applications into SocioScope.

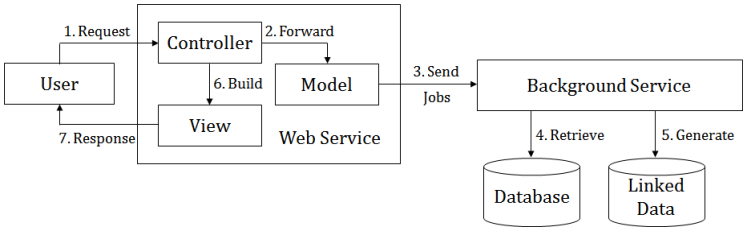


Fig. 1. Overview of SocioScope.

Dealing with big data is one of big challenges of SocioScope. Hence, we consider applying big data processing techniques when we design the system. MongoDB database is selected because it is schema-less (i.e., easily extending and altering extra fields), NoSQL, fast access (i.e., using internal memory for saving working set in order to allow faster data access). In addition, MongoDB can be used together with Hadoop to power big data system.

Besides, we also build a Linked Data (LD) database for better understanding and generating structured data. When a text data is collected, we tokenize this sentence to create a bag-of-words. Our system will retrieve information on Linked Open Data sets (i.e., DBpedia and Freebase) based on words. We use LD database for producing specific metadata of our document and for conducting context-based analysis as well.

2.2 Modules

All modules which are described here are supported by our system. Besides, we also mention about URL schemes in order to directly access these modules. The URL must follow format: "domain/subpage". Belows are list of all modules in which SocialScope contains:

User Management

- Sign up (/sign_up): User must register an account on SocioScope for accessing the system. This URL shows a form to apply for a new account. There are some script for verifying content in which user inputs.
- Sign in (/sign_in): This page is for an individual to gain access by passing his username and password. After successful logging in, user is redirected into SocioScope homepage.
- Home (/home): This is the homepage of SocioScope. All the information about the system is shown here.

Social Data Collection

- API account (/api_account): In order to process authorized requests to obtain data from sources, we have to register for applications or tokens from API. This page is used for managing all the API accounts.

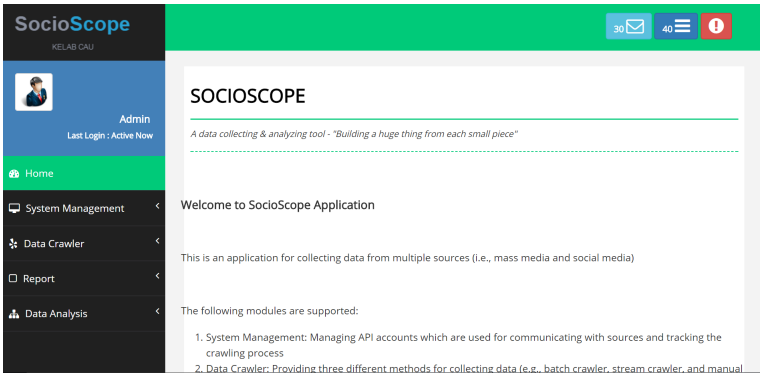
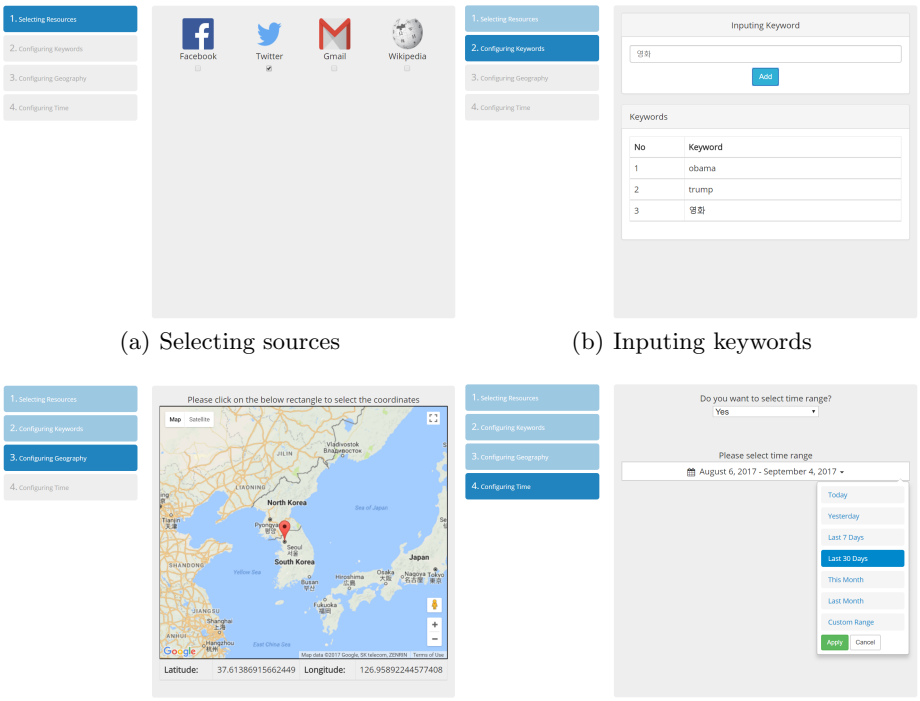


Fig. 2. The home page of SocioScope.

- API tracking (/API_tracking): Each API has different rate limit strategies. If the number of requests overcome limitation, API will return error code and no data is acquired. We can control API rate limits by using this page.
- Batch crawler (/batch_crawler): Batch crawler is used when you want to deal with blocks of data that have already been stored over a period of time. It works well in case user wants to process large volumes of data. We supports crawling data not only by using keyword but also by using location and time. Tab. 3 shows the user interface of crawler page. Moreover, we also focus on solving multilingual problem. User can collect data with different languages.
- Stream crawler (/stream_crawler): The main features of stream crawler is similar with batch crawler. However, stream crawler is used when you want to analyze data in real time. Stream crawler is useful for fraud detection tasks (e.g., in healthcare, telecommunications, or banking area). We create the set of listeners for collecting data. Every time a new data from data sources is generated, it is automatically collected by these listeners.
- Manual crawler (/manual_crawler): Almost data sources use ID for managing data. In case user has already had list of data ID, we support this feature for getting data by passing ID into system.
- Collected data (/collected_data): After crawling, data is stored in database. This page allows retrieving all the data from database. Pagination techniques are used for breaking large data into smaller portions to increase system performance. In addition, we support extracting data to file (e.g., text and csv file) using json format.

Social Information Generation

- Natural Language Processing (/nlp): Providing basic tools for natural language processing (e.g., tokenizer, POS tagger, named entity recognizer, stemming, lemmatization, sentiment analysis, and coreference resolution). This is an important step for understanding text data.



(a) Selecting sources (b) Inputing keywords

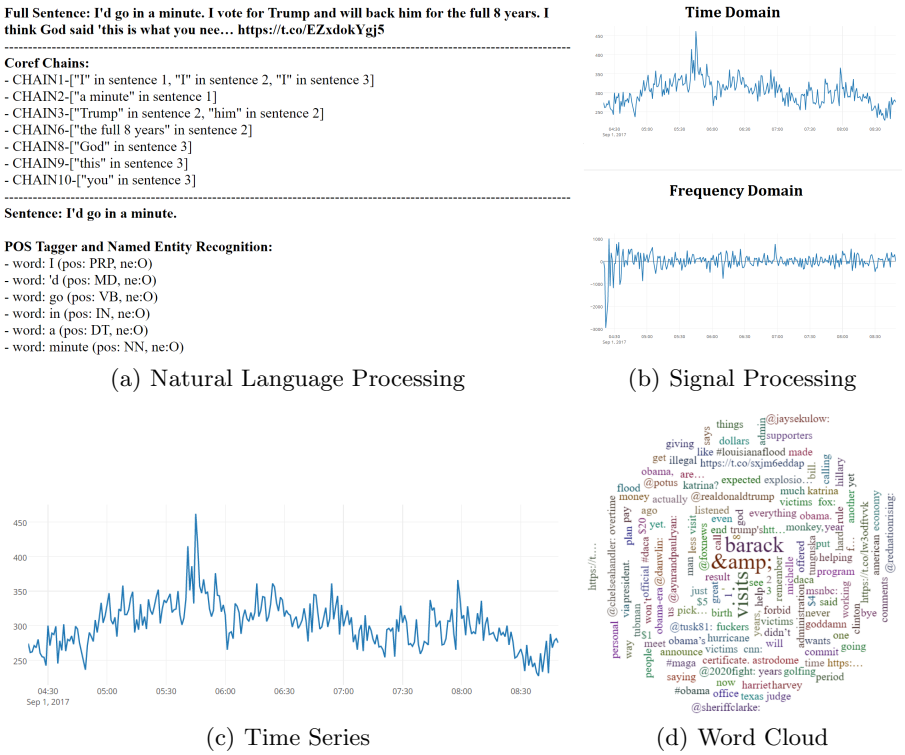
(c) Choosing location (d) Picking up time

Fig. 3. The crawler feature.

- Time Series (/time_series): Displaying how the frequency of data change over a period of time. From the time series visualization, we can easily recognize patterns of data or even observe abnormal signal.
- Signal Processing (/signal_processing): Converting signal from time domain to frequency domain by using different techniques (e.g., fast fourier transform, discrete fourier transform, discrete cosine transform, discrete sine transform, discrete hartley transform, fast wavelet transform, and wavelet packet transform). Because of many proved functions, it is easier to compute and process signal in the frequency domain rather than in the time domain.
- Word Cloud (/word_cloud): Word cloud is a visualization method to display words which appear more frequently in the source text in a prominent way. It helps us to generate a starting point for deeper analysis later on (e.g., judging that there are words which are co-occurrence in specific topics) [3]

3 System Performance

In this section, we demonstrate the performance of SocioScope. There are three issues that we want to prove which are: *i*) the performance of crawling process,



(a) Natural Language Processing (b) Signal Processing

(c) Time Series (d) Word Cloud

Fig. 4. Social information generation features

ii) the effectiveness of MongoDB when dealing with bid data, and *iii*) the time consuming for conducting data analysis tasks. All the experiments are conducted by using a computer with specifications as follows: Intel(R) Core(TM) i5-4590 CPU 3.30GHz with 12GB RAM. Twitter is selected as the source for the experiments because we can consider data on this source as big data. Moreover, the request limitation of Twitter API (180 calls every 15 minutes) is also the challenge that we want to overcome.

We first measure the performance of crawling feature by collecting data in 1 minute, 10 minutes, and 100 minutes by using batch crawling and streaming crawling features. In order to solve the request limitation problem, we create 35 different Twitter API applications and build an application pool. An application pool is a collection of Twitter API applications that work based on queueing theory. Every time an application reaches limitation, it is automatically moved to end of queue and another available application is used. Therefore, the crawling process is conducted continuously without any delaying. Moreover, we also implement thread pool for parallel crawling. At any time, a thread will be used as long as it is still available. Obama and Trump are two keywords that we choose for testing the batch crawler and stream crawler respectively. The result

in Tab. 1 shows that batch crawler is quite stable (about 4200 tweets per minute) while stream crawler is quite dynamics due to the real-time data property. From the result, we draw the conclusion that by using SocioScope we can collect much more data than using HTTP requests as normal.

Table 1. Crawling process performance

Time	SocioScope Crawler		HTTP Requests
	Batch Crawler	Stream Crawler	
1 minute	4,205 tweets	3,529 tweets	600 tweets
10 minutes	43,898 tweets	28,190 tweets	1,800 tweets
100 minutes	459,796 tweets	310,627 tweets	10,800 tweets

Further, we interested in proving the effectiveness of MongoDB database when dealing with big data. In order to optimize system performance, we first apply indexing technique on frequent retrieving record to minimize the number of disk accesses required. Besides, WiredTiger storage engine is also applied. This engine supports of document level locking. Therefore, system archives better concurrency. Furthermore, WiredTiger uses snappy compression algorithm to reduce the number of data which have to be written or read from the disk. Fig. 5 shows that NoSQL is better than SQL databases when dealing with big data. The time is measured when user clicks on the retrieve button until the result is shown.

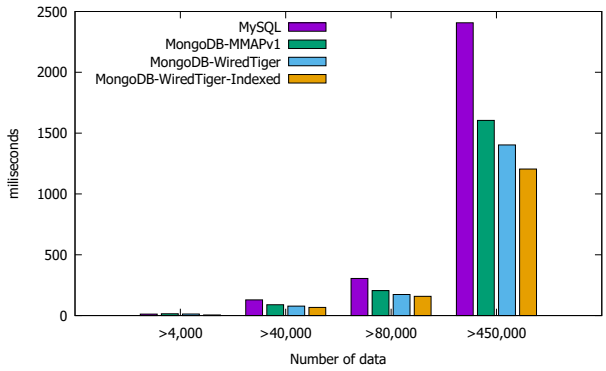


Fig. 5. Database performance comparison.

Finally, we focus on data analyzing tasks. Time consuming for natural language processing, signal processing, time series and word cloud visualization tasks are approximately with the result in Fig. 5. This shows that the time for analyzing data of SocioScope is insignificant.

4 Conclusion

In this paper, we introduce about SocioScope which is a framework for collecting and analyzing social data to create social information. There are various analyzing tools (e.g., natural language processing, signal processing, visualization) which are supported to discover our society from social data. In addition, the output of SocioScope can be taken advantages for generating social knowledge (e.g., event detection [5], trust-based recommendation system [4]), or even social wisdom.

We also plan some future work for enhancing the utility of our system. Some other sources will be investigated for integrating into our system such as mass media source (e.g., newspaper and radio), social media sources (e.g., Instagram, Flickr, and Foursquare), and sensor source (e.g., camera and wearable devices) to enrich social data. Besides, we consider applying Hadoop, which is a powerful framework for dealing with big data, to improve our system. Finally, other analyzing tools (e.g., sampling, transformation, denoising, and feature extraction modules) will be applied for better understanding social data.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2017R1A2B4010774).

References

1. Ackoff, R.L.: From data to wisdom. Journal of applied systems analysis 16(1), 3–9 (1989)
2. Erevelles, S., Fukawa, N., Swayne, L.: Big data consumer analytics and the transformation of marketing. Journal of Business Research 69(2), 897–904 (2016)
3. Heimerl, F., Lohmann, S., Lange, S., Ertl, T.: Word cloud explorer: Text analytics based on word clouds. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS 2014), Hilton Waikoloa, Hawaii, USA, Jan 6-9, 2014. pp. 1833–1842. IEEE (2014)
4. Hoang Long, N., O-Joun, L., Jai E, J., Jaehwa, Park amd Tai-Won, U., Hyun-Woo, L.: Event-driven trust refreshment on ambient services. IEEE Access 5, 4664–4670 (2017)
5. Nguyen, D.T., Jung, J.J.: Real-time event detection on social data stream. Mobile Networks and Applications 20(4), 475–486 (2015)
6. Smith, E.A.: The role of tacit and explicit knowledge in the workplace. Journal of knowledge Management 5(4), 311–321 (2001)