Multimed Tools Appl (2017) 76:10893–10916
DOI 10.1007/s11042-016-4038-2

City digital pulse: a cloud based heterogeneous data analysis platform

Zhongli Li[1] · Shiai Zhu[1] · Huiwen Hong[1] · Yuanyuan Li[1] · Abdulmotaleb El Saddik[1]

Received: 20 March 2016 / Revised: 26 September 2016 / Accepted: 4 October 2016/
Published online: 11 November 2016

Abstract In recent years, increasing attention has been paid to developing exceptional technologies for efficiently processing massive collection of data. This is essential in the research on smart city, which involves various types of data generated by different kinds of sensors (hard and soft). In this paper, we propose a cloud-based platform named City Digital Pulse (CDP), where a unified mechanism and extensible architecture are provided to facilitate the various aspects in big data analysis, ranging from data acquisition to data visualization. We instantiate the proposed system using multi-model data collected from two social networks, namely Twitter and Instagram, which can provide instant geo-tagged data. Data analysis is performed to detect human affections from user uploaded content. The information revealed from the collected social data can be visualized at multiple dimensions through a well-designed Web application. This allows users to easily sense changes in human affective status and identify the underlying reasons. This offers priceless opportunities to improve the decision making in many critical tasks using the detected attitudes in the social messages, such as promotion strategy for companies or new policy making for the government. Our experiment results confirm the effectiveness of the proposed architecture and algorithms.

✉ Shiai Zhu
zshiai@gmail.com

Zhongli Li
lzl19920403@gmail.com

Huiwen Hong
huiwen.hong@gmail.com

Yuanyuan Li
yuanyuanli199@gmail.com

Abdulmotaleb El Saddik
elsaddik@uottawa.ca

[1] Multimedia Computing Research Laboratories (MCRLab), University of Ottawa, Ottawa, ON, Canada

Keywords Smart city · Cloud-based system · Social media · Data analytics · Data visualization

1 Introduction

We have entered the era of big data [3]. An incredible amount of data is generated daily by "hard" sensors that come with our personal electronic devices or pre-installed in the facilities around us. Nowhere is this more apparent than a smart city [23] which deploys a wide range of smart objects built with various kinds of sensing technology. This opens up the opportunity for different kinds of sensing applications to become parts and parcels of modern living. For example, smart cars with lidar sensors can sense traffic build up on the street and then devise intelligent plan to collaboratively mitigate it [38]. In addition to hard sensors, Web data (e.g., social networks) is another rich knowledge source, that captures data produced by users themselves that comes with auxiliary data such as geo-location and social data. In this respect, websites and online social networks on the web can be regarded as some kind of "soft" sensors. Performing data analysis on these big data can bring about useful insights that can help elevate the quality of lives of fellow citizens and improve the management of the city.

As a huge number of sensors produces more data for consumption, this brings about new challenges to discover useful multi-faceted information from the massive pool of data. Unfortunately, the heterogeneity of sensors, data and employed algorithms causes difficulty in two respect. Firstly, when different types of sensors are used, integration becomes a complex issue that poses many challenges to system design and implementation. For example, different sensors may come with different interfaces, programming languages and operating systems. This may result in different naming conventions of the metadata for the same concept. Furthermore, some sensors do not work independently and needs to be processed in conjunction with other sensors. This creates an increase in demand for data conversion and reliable communication between devices. Secondly, the sheer size of the data to be processed poses challenges on data storage, retrieval and analysis algorithms which are usually computationally expensive. Thus a well-designed architecture is needed.

Recently, with the emerging of cloud-computing technology [27], it is convenient to deploy and operate a cloud-based system. Many applications [17, 38, 44, 46] are developed under the paradigm of cloud-computing. While development of a powerful cloud-computing platform is too expensive for many companies and institutions, some commercial cloud services (e.g., Amazon Cloud) can be employed to economically develop cloud-based systems. However, previous works are almost about a specific application, and thus not extensible. The scenario that our work tackles in this paper is far more complex where smart sensing applications essentially "talk" to each other all the time to exchange information which in turn are used to make more informed and smarter decisions. The proposed system, namely City Digital Pulse (CDP), is designed to tackle the whole data science pipeline, from data collection to visualization. For the system design, we carefully divide the architecture into several functionally independent components. Communication between components are achieved by APIs developed within each component. In this way, the system becomes more reliable and can be easily extended to include other applications by adding the corresponding APIs.

In order to demonstrate the feasibility of our proposed architecture and design requirements, we implement the platform that processes social data. Social data is highly dynamic where huge streams of data is produced on a constant basis by different users on a wide

range of topics. It is also heterogeneous where data comes in many different forms, both structured and unstructured. The complexity of the environment is the main reason why it is selected to evaluate the performance of our platform. Specifically, we focus on the application of detecting sentiments in the geo-tagged messages posted in social media. Different from the works in [4, 14, 21], which only focus on aspects of the data science pipeline(i.e., visualization [21] or data collecting [4]), we introduce a integrated architecture with optimized solutions on data acquisition, storage, retrieval, process and visualization. Similar functions are achieved in the system proposed in [48]. However, their work focuses mainly on topic mining for the data analysis component. In contrast, our work comprehensively covers all major components in the data science pipeline both from the theoretical and applications perspective. In [43], a cloud based tool is developed to monitor the clients' attitudes toward brands according to discussions in social media. Compared to the work in [43] which also implements many functional components, our developed system follows several design principles (i.e., extensibility). This allows each component and the overall architecture to be highly adaptive to the evolving needs of users. In [39], data analysis is performed to explore the citizens' reactions on new policies released by the government. Our work is different in that the public can freely access the proposed system deployed on both our local server[1] and commercial cloud server.[2] In addition, our system is more flexible where it can be customized to satisfy the needs of different individual customers.

The remaining sections are organized as follows. Section 2 briefly reviews the related works. The design requirements are summarized in Section 3. In Section 4, each component in the framework of our system and the implementation details are illustrated. Finally, Section 5 concludes this work and gives the possible future works.

2 Related works

Our system is developed in the context of smart city, which is defined by IBM as the use of information and communication technology to sense, analyze and integrate the key information of core systems in running cities [12]. In order to achieve intelligent responses, various sensors are deployed in the city to collect different kinds of data related to our daily livelihood, environment, public safety and etc [25, 47]. Because of the great research progress on smart city, dynamic scalability can be supported in many applications [31, 40]. In [9], cities' sustainability is improved by using the data mining techniques. In [5], the existing street lighting infrastructure is enhanced with additional sensors and control to achieve smart lighting. In [19], a sustainable intelligent transportation system is designed for energy conservation in the city. In [24], a cloud based system of Internet of thing is presented to achieve smart services for end users. All these works rely on different kinds of hard sensors deployed in the city. Thus, previous works are most at theoretical or conceptual level.

The rising of cloud-computing and cloud storage offers a great opportunity for data intensive applications, such as the smart video surveillance [11] and house data analysis [45]. In the cloud-based system, smart objects are deployed at the bottom layer (hardware), and applications are defined at the top layer. A cloud-enabled middle layer (services) defines different interfaces (e.g., software-as-a-service) to deliver functional values. In [23], the

[1] http://citypulse1.site.uottawa.ca.

[2] http://citydigitalpulse.us-west-2.elasticbeanstalk.com.

authors provide a theoretical perspective to deal with the big data in smart city and address the issues of sustainability and socioeconomic growth of the city. In [26], the data storage and transmission among different kinds of data generated in smart city is implemented in a distributed architecture. The advantage is demonstrated through several experiments. In [42], a framework is proposed to provide scalable and real-time IoT services by organizing the services in a tree structure. In [32], a hybrid cloud architecture is presented for supporting coordinated emergency management and first responder localization. The management, computing and storage are integrated in the system. In general, cloud computing techniques have been applied successfully in a variety of applications. Hence, we adopt the same strategy and use commercial cloud services to deploy our system.

In the research of smart city, one important issue is data analysis. In [3], smart city data analysis is contrasted against general big data analysis. One key distinctive features of smart city analysis is that data analysis is only performed on demand and the data handled is typically either semi structured or unstructured. Most of the works are devoted on processing the data from hard sensors, such as [18], where real-time event processing and clustering algorithms are proposed using the intelligent servers from the OpenIoT project. We argue that smart city should also involve the data generated by soft sensors (e.g., social media). Current works on social media differ from each other in term of data types, social network platforms, types of applications and analysis techniques. For example, sentiment analysis can be performed on text messages [29, 33], social images [7, 8, 16] or even user networks [20, 41]. The proposed algorithms may employ pre-defined knowledge sources or supervised learning techniques. Furthermore, different approaches are proposed to handle the data from different kinds of resources, such as the stereotyped data from Internet-based retailers [13] and free-formed Twitter messages [20]. The usefulness of web data has been demonstrated in many areas such as business intelligence and analytics [6] and crises detection in social media [28]. Big data analysis is not trivial, the challenges issues are summarized in [1]. In the previous works, the social data analysis is treated as an separate research problem, and rarely considered in the context of smart city. In this paper, we will focus on a framework for smart city that considers both hard and soft sensors.

3 Design requirements

We target at a unified platform, which integrates diverse information generated in a smart city, and provide good experiences for general users. Thus, there are some design requirements need to be followed. In this section, we summarize them as follows.

1. Data safety: The system design should consider backup solutions to handle unexpected events such as system crash or unauthorized access. A disaster recovery plan should be put in place to mitigate any contingency scenarios.
2. High reliability: The system includes many components, and each component may have several sub-systems. To improve reliability, the sub-systems should be loosely coupled from each other so that any updates or failures in any one sub-system would not adversely affect the operation of the system as a whole. The system should also be able to automatically detect the errors or possible risks, and make the appropriate remedies.
3. High scalability: Scalability refers to the capability and potential of a system to handle workload growth. This is especially important to our system, as we are working on an unified platform for the smart city, which will process large databases and run various resource-consuming algorithms.

4. High extensibility: Extensibility is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Extensibility is important to support future growth such as installing new types of sensors or expanding the array of product offerings.
5. High efficiency: Our designed system can be used on different kinds of end devices (e.g., laptop and mobile phone). As some devices are inherently resource limited and the wireless transition is expensive, resource management strategy should achieve high efficiency in terms of computational cost, communication cost and storage. Most of the computing is conducted on Web servers. End devices may maintain some critical information and light-weight services for visualization and interaction. Thus the data transfer between servers and client can be reduced.
6. User friendliness: The user interface and system design should deliver high quality user experience. The system should provide the most valuable information in real-time. In addition, the user interface should provide an optimal view and operation experience for all kinds of end devices that may come with all different screen sizes.

4 City digital pulse (CDP) platform

4.1 System architecture

The proposed City Digital Pulse (CDP) is an end-to-end architecture to facilitate the various aspects in big data analysis, ranging from the data acquisition to the data visualization. As showed in Fig. 1, CDP consists of three conceptual parts: sensors, cloud service provider and end devices. In addition, each part may include several sub-systems to meet the requirements of different tasks. Sensors can be either hard sensors (e.g. GPS) or soft sensors (e.g., social media), which are responsible for generating data. The collected raw data can be further processed and stored on the cloud side, where data transition and retrieval are also deployed. In order to reduce the cost of communication and computing resources in end devices, we deploy most of the computational or storage intensive works on the cloud. The processed information is then delivered to the user through a web-based application.

For proof of concept, we implemented the architecture for an application, which aims at detecting and visualizing citizens' sentiment from the social data. Figure 2 shows the major components and their connections in the architecture. Each part is developed separately, and

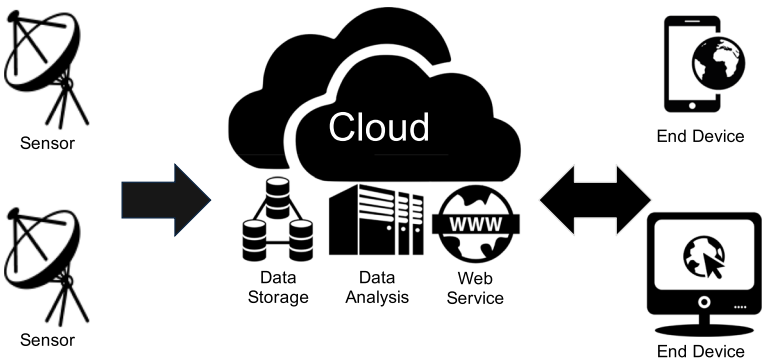


Fig. 1 System architecture of city digital pulse (CDP)

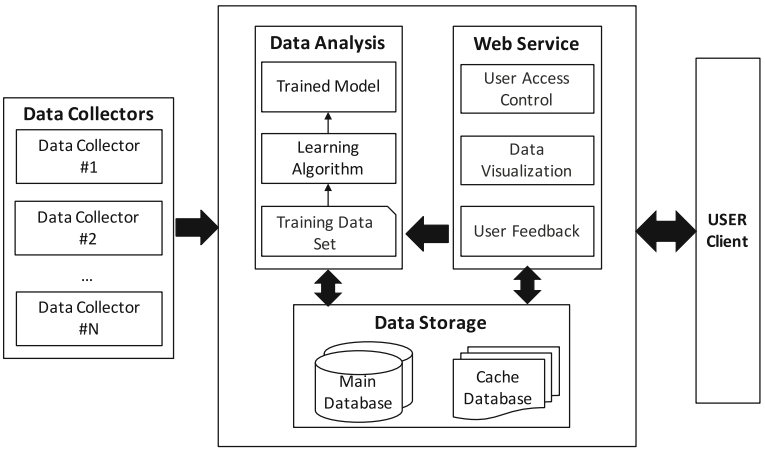


Fig. 2 The major components and their functional relationships in the city digital pulse (CDP)

designed as APIs with a unified interface. Thus the overall system can be easily extended. The prototype comprises 5 major functional components:

- **Data collector** is used to acquire and upload data. There can be multiple collectors for different data sources. In CDP, we develop two collectors for gathering data from two social networks (i.e., Twitter and Instagram). More details of the data collector will be introduced in Section 4.2.
- **Data storage** is responsible for structuring and storing various kinds of data in CDP. To optimize the security, scalability and retrieval speed, we design two major databases for storing different kinds of data: cache database and main database. The details will be further illustrated in Section 4.3.
- **Data analysis** consists of a set of algorithms, which are used for filtering data, representing data and mining useful knowledge from the data. The adopted solutions are determined by the data types and applications. In our system, sentiment analysis is modeled as a classification problem, which involves data collection, feature extraction, classifier learning, prediction and information aggregation. We will provide the details for each of them in Section 4.4.
- **Web service** provides an interface to show the information explored from the data (e.g., visualizing sentiments on the map) and receive user feedback (e.g., allowing user to annotate on the training data). Other functions include access control and adaptation to different end devices. Section 4.5 illustrate our designed services in CDP.
- **Cloud service** aims at integrating the aforementioned components on the cloud to achieve high scalability in data processing capability and high Quality-of-Service (QoS) for end users. In Section 4.6 we will describe the implementation details and the system deployment on the cloud.

4.2 Data collection

As the data may be contributed by various sensors, we define a collector for each sensor separately. The scheduling is controlled by a task manager, which also monitors the collector status and responses to user inputs. Figure 3 shows the structure of the collector.

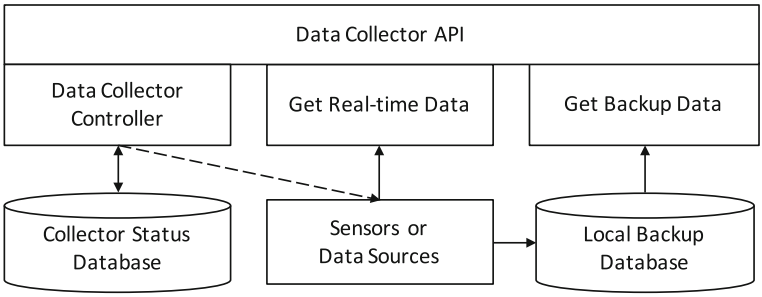


Fig. 3 Structure of our designed collector component

The data collector provides three kinds of APIs. It can initiate or stop data collection based on system load and user input. It can also update the status of running tasks in the collector status database. For example, when a new sensor is added, the controller will estimate the workload and split the collecting process into several tasks. Control signal is sent to the corresponding collection API. The process is dynamically updated in our system. In addition, abnormal situations such as crashed task is handled by the controller and recorded in the database. We first define an API to collect incoming data based on the user inputs. As the system on the cloud may crash and the API may fail to receive data from sensors, we further define a local backup database to temporally store the data from sensors. Another API will be activated to complete the missing data in the event of a system crash.

4.2.1 Twitter message collector

Twitter messages are collected by using the Twitter streaming API, which can be used for gathering real-time messages. We target Geo-tagged messages.

The Twitter streaming API allows listening to be performed within an area defined by a bounding box (rectangle) on the map. The Geo-locations of the two corner points are parameters. As the boundary of a city is usually irregular, we employ multiple rectangles to mark the boundary of a given city. The number of returned messages in one listening area is limited by Twitter. Thus, there may be missed messages for large area, which may include messages more than the limitation. On the other hand, small area results in more rectangles, and this in turn means more subtasks need to be created, leading to higher resource usage. In our experiments, we found that diagonal of 60KM is a suitable choice for the rectangle to hit a balance between data completeness and system efficiency. Figure 4 shows an example of the adopted split for the area of Ottawa city, where four sub-tasks are created. Note that the overall listening area exceeds the boundary of the targeted city. Thus the received messages are filtered in the message upload service based on their attached Geo-locations.

4.2.2 Instagram message collector

Instagram posts are collected using the Location Endpoints Search API. Unlike Twitter streaming API, the Instagram API only allows developers to search for Instagram posts in a circle with radius 10KM. In order to get complete messages of a city, we split the city into several circles and assign a task for each circle. Note that the circles are defined to be overlapped. Duplicated messages collected from the overlapped circles are removed according to their unique IDs assigned by the Instagram API. The Instagram API has two

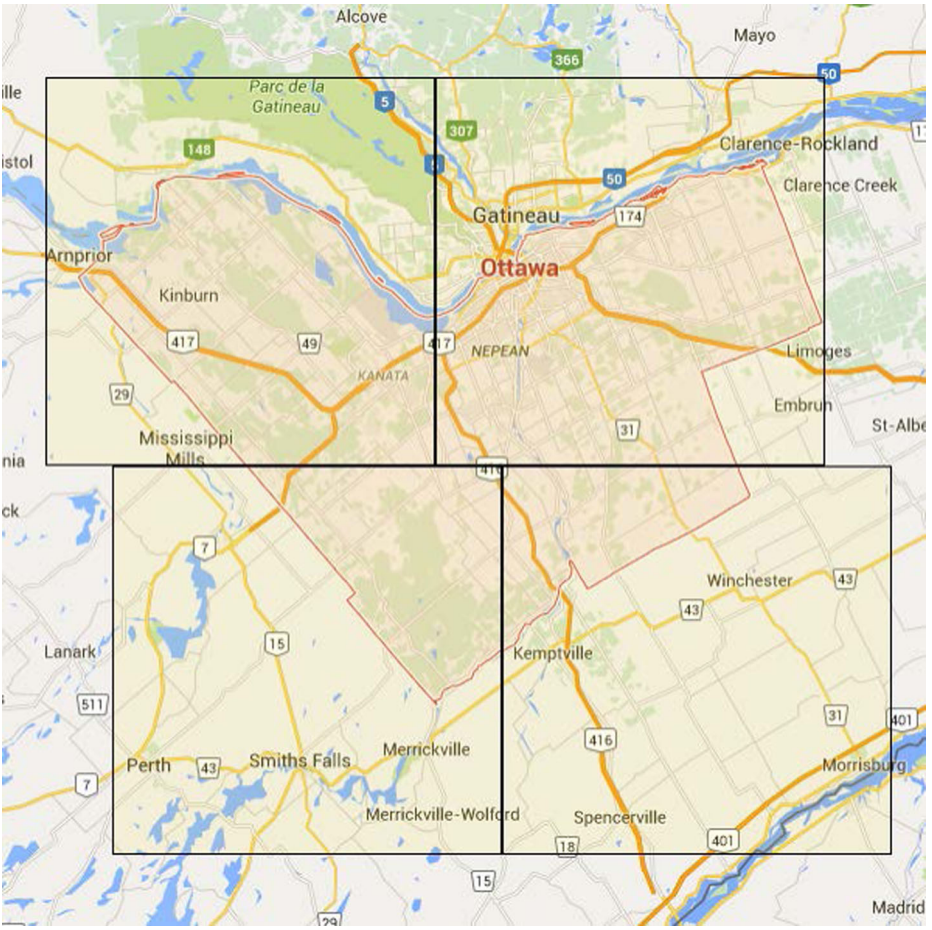


Fig. 4 The four sub-regions for collecting data of Ottawa

parameters. One is used to define the search area, the other is the time period. This feature makes it easier for us to get historical data during a specific time period. In order to collect the real-time posts, we set the start timestamp a few minutes before the current time and the end timestamp as current time. Both the start and end timestamps is increased automatically to continually get posts. This will result in a delay of few minutes, which is, nevertheless, acceptable in real-world applications.

4.3 Data restructuring, storage and retrieval

4.3.1 Data restructuring

The format of collected raw data varies across different data sources. For example, most of the data collected from the social media is in JSON (**J**ava**S**cript **O**bject **N**otation) format which can have diverse data fields. We can utilize NoSQL database (e.g., MongoDB) to store unstructured data. However, this may result in high storage cost and inefficient data

retrieval, which are undesirable in big data analysis. Therefore, we filter the raw data according to the requirements of the applications and store structured data in SQL database. In our prototype, we extract text, image URL, geolocation and timestamp from social messages and store them in the SQL database.

4.3.2 Database design

The designed databases and corresponding operations are showed in Fig. 5. As data security is one of the most important aspects in our system, we use the concept of Master Slave Replication to protect the data. Once there is a change of data in the master database, a change log will be sent to every slave database to update the data accordingly. If the master database crashes, the function of master database will be replaced by one of the slave databases. As data collection is performed continuously, there will be frequent read and write operations in the system. In order to improve the response speed, we split the read and write operation by utilizing the master and slave databases.

When the amount of data generated from the sensors increases, after a certain point, keeping all data in a single database may result in significant drop in access time. For example, collecting data from 30 cities in Canada, our database reached over 3 million rows just over 20 days after the launch of our system. To address this problem, we split the large database into several small databases. There are two kinds of partitioning schemes. Vertical partitioning is used to split the column of the data table according to the meanings of different data fields. In this way, access to the needed data fields will be improved. Figure 6 shows an example of database partitioning. The original database is split into city detail database and record data database. The left one only keeps information related to cities, such as name and country. The right one only keep information related to messages. Many repeated records of city in the original database are removed, and thus the storage space can be saved. On the other hand, Horizontal partitioning (Sharding) splits the data by rows.

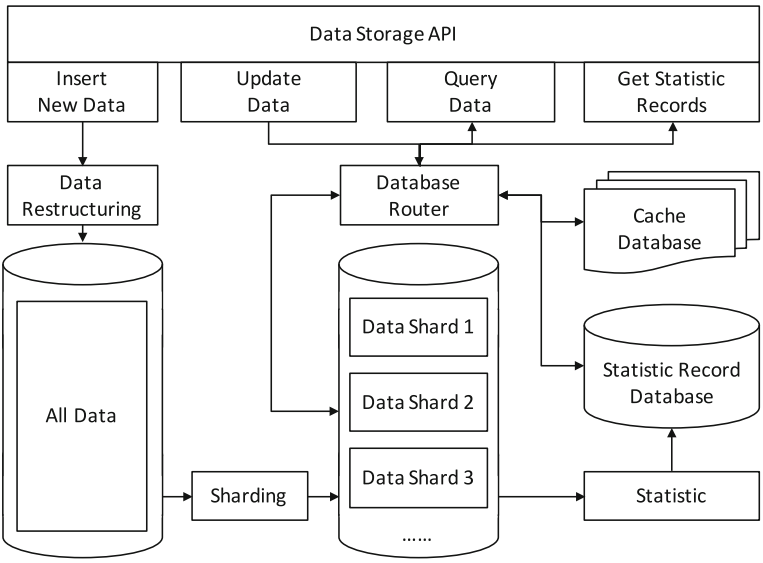


Fig. 5 Design of the data storage component

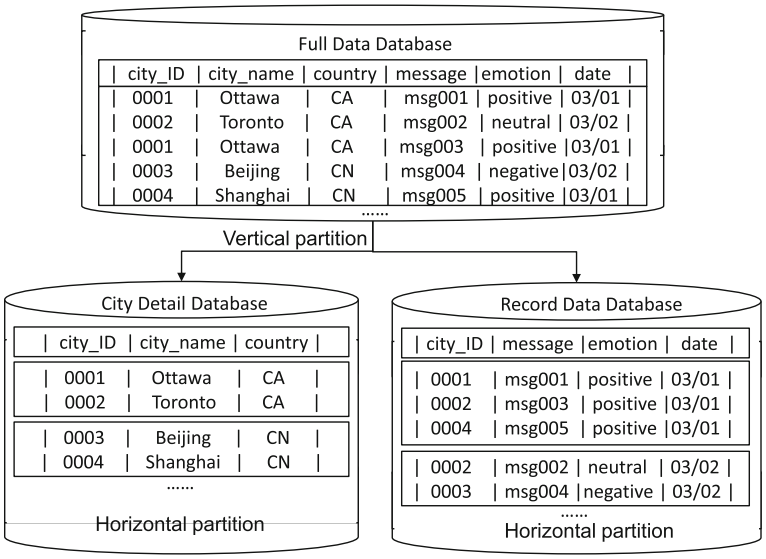


Fig. 6 Vertical and Horizontal partition of database

Each sub-database includes data having contiguous values in the key field. This is especially helpful for querying a range of data. For example, we split the items in the right database in Fig. 6 by uploaded date, since it is a common practice to refine a query using a specified time period as filter condition. It is easier to locate the targeted messages in the filtered list. The target messages can be easily located according to the start and end timestamp. The query of different small databases is controlled by the database router which maintains a Shard key for each sub-database (Shard). Note that partitioned database is not efficient for handling complex queries, which require merging the results from different sub-databases. However, the effect on the overall query performance is marginal, since complex queries are rare and the number of corresponding messages is small.

To improve query speed, we can add indices into the data table when appropriate. However, this will incur additional storage space especially when all fields are indexed. In our implementation, we only index three fields, namely timestamp, location and keyword, which are frequently used in queries generated by some applications.

4.3.3 Query optimization and caching

A query is a set of conditions defined on the data fields. Retrieval can be achieved efficiently by setting the start and end index of the timestamp. However, the same strategy is not suitable for location as the boundary is irregular. In order to avoid matching messages one by one, we first minimize the search area using the minimum bounding rectangle of the target city. Messages in the rectangle can be located efficiently in the database by simply setting the start and end indices of the latitude and longitude. The extracted messages is then stored in the memory.

To store massive amount of data in the system, we use a hard disk database as our main database. However, the speed of hard disk is much slower than the memory. Caching frequently used data into the memory can speed up the query and improve the quality of user

experience. As showed in Fig. 5, all incoming queries will be sent to the database router. If same query has been performed before and the data is in the cache database, the data will be returned immediately without querying the main database. For new queries, the router will query the main database and return the data to both the user and cache database. When the system runs out of memory space, the data corresponding to infrequent queries will be replaced.

4.4 Data analysis

The system can be easily extended to handle different types of applications, each requiring different types of data analysis techniques. In this section, we utilize sentiment analysis of social data as an example to explain the data analysis component. Both the approaches and experimental results are presented in this section.

4.4.1 Sentiment analysis method

Extensive research has been devoted to the study of sentiment analysis of different media types such as textual documents and visual images. In social media, plenty of opinion-rich textual data is posted to voice users' opinion on different topics. State-of-the-art approaches on sentiment analysis can be roughly divided into two groups: Lexicon-based and statistical learning methods.

Lexicon-based approaches adopt the lexicons consisting of a set of emotional words, which are assigned with scores to indicate the strength level of sentiment. Two widely used sentiment lexicons are SentiWordnet[3] and SentiStrength.[4] SentiWordnet is constructed based on the WordNet,[5] where words with the same meaning are grouped into a synset. In SentiWordnet, each synset is associated with three numerical scores corresponding to its level of positive, negative and objective sentiment. As one word may belong to multiple synsets, SentiWordnet further defines which one is the first meaning, second meaning and so on. Given a word, the total sentiment score is computed by weighting its synset scores based on the rank of meanings. As showed in Fig. 7, adjective word "cute" receives positive score 0.54 in SentiWordnet.

In SentiStrength, the positive strength of a word is defined as a number between 1 and 5. The meaning of 1 is not positive and 5 means extremely positive. Similarly, the negative strength score is between -1 and -5 indicating the sentiment ranging from "not negative" to "overly negative". In the example of Fig. 7, positive words include cute, adorable and romantic etc.

Given a document including N individual words, the overall sentiment score can be computed by averaging over all the words. Another way adopted by SentiStrength is to compute the sum of the maximum positive score and minimum negative score. By using the lexicon-based approaches, the message showed in Fig. 7 is labeled as positive sentiment using either SentiWordnet or SentiStrength.

In the statistical learning approach, sentiment analysis is posed as a standard classification problem which includes two major components: feature representation and model

[3] http://sentiwordnet.isti.cnr.it/.

[4] http://sentistrength.wlv.ac.uk/.

[5] https://wordnet.princeton.edu/.

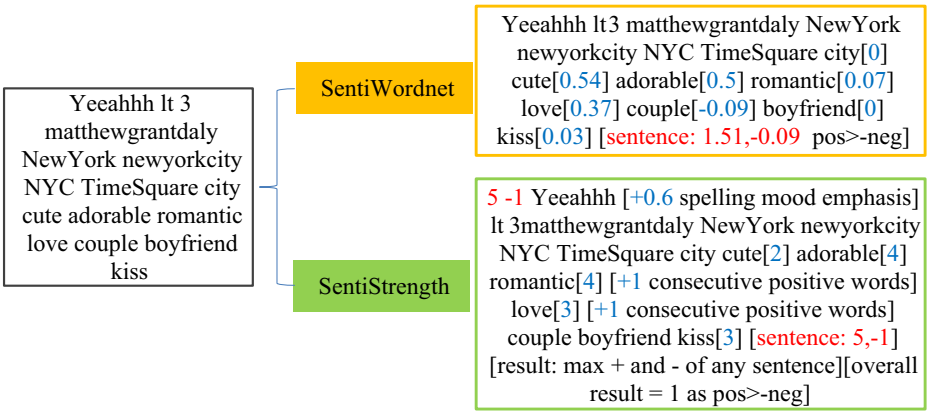


Fig. 7 Sentiment analysis by using two Lexicons: SentiWordnet and SentiStrength. The contents within "[ ]" is the analysis results. *Blue* value means the score for each individual word, and *red* ones are the overall results of the sentence

learning. In this paper, we use the standard textual feature (i.e., Bag-of-Words) and linguistic feature designed for affective computing. BoW feature represents the word distribution in a document. Each element in the feature vector corresponds to a word from a pre-defined vocabulary $\mathcal{V} = \{w_1, w_2, \ldots, w_n\}$. The value can either be a binary value indicating the appearance of the corresponding term or a counting value indicating its term frequency (TF).

Different from the BoW feature, linguistic feature considers the sentimental symbols embedded in the documents. In this paper, we experiment the linguistic features proposed in [35], such as the number of exclamation marks "!" in the message.

For the learning algorithm, the approaches widely used in previous works [10, 29, 34] on text sentiment analysis are Naive Bayes (NB), Support Vector Machines (SVM) and Maximum Entropy (ME). However, there is no conclusion on which one is the best, and the choice is variant across different domains. In this paper, we adopt SVM which has been demonstrated to be effective in different applications [2, 15, 22]. Both LinearSVM and KernelSVM using the polynomial kernel will be evaluated in the following section.

4.4.2 Empirical study

In this section, we conduct experiments on SemEval [36] and MVSA [30] datasets to evaluate the different approaches. SemEval is a large dataset consisting manually labeled tweets constructed for an annually organized competition.[6] MVSA is constructed for multi-view sentiment analysis. Two subsets are provided in MVSA. We select the MVSA-multiple dataset, where each tweet is annotated by three annotators. Only messages receiving at least two same labels are selected in our experiments. In this way, the annotations are expected to be more accurate. Table 1 lists the details of the two datasets.

The performance is evaluated by accuracy and F-score. Formally, accuracy is defined as

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{1}$$

[6] http://alt.qcri.org/semeval2016/.

Table 1 Statistics of the two datasets

Dataset	#positive	#neutral	#negative	#all
MVSA	14,089	2,684	1,851	18,624
SemEval	2,279	3,049	851	6,179

where tp, tn, fp and fn indicate true positive, true negative, false positive and false negative respectively. F-score is defined as

$$F\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{2}$$

where precision and recall are calculated by

$$\begin{aligned} precision &= \frac{tp}{tp + fp} \\ recall &= \frac{tp}{tp + fn} \end{aligned} \tag{3}$$

In our experiment, F-score is computed on positive sentiment (F-positive), negative sentiment (F-negative) and neutral sentiment (F-neutral) respectively, and their average is denoted as F-average.

For each dataset, we randomly split the data into 10 groups, where 9 groups are used for model training and testing is performed on the other group. Table 2 lists the average performance of the results for each studied method. Generally, the statistical learning approaches perform better than the lexicon-based approaches. This is because that the messages contributed by uncontrolled users are short, noisy and incomplete. In other words, sentiment-related clues are weak and difficult to be detected by using lexicons owing to the limited number of emotional words found in social messages. Thus sentiment analysis on tweets needs more advanced approaches and dedicated designs. We can see that the F-positive of statistical learning approach is much higher than F-negative and F-neutral on

Table 2 Performance comparison of different approaches on SemEval and MVSA datasets. The best results are highlighted

	Accuracy	F-positive	F-neutral	F-negative	F-average
			SemEval		
LinearSVM+BoW	**0.680**	**0.684**	0.740	0.364	**0.596**
LinearSVM+Ling.	0.675	0.657	**0.742**	0.374	0.591
KernelSVM+BoW	0.655	0.662	0.718	0.369	0.583
KernelSVM+Ling.	0.668	0.650	0.735	0.351	0.578
SentiWordnet	0.551	0.541	0.587	0.371	0.499
SentiStrength	0.562	0.563	0.601	**0.384**	0.516
			MVSA		
LinearSVM+BoW	0.752	0.861	0.076	0.265	**0.401**
LinearSVM+Ling.	**0.760**	**0.863**	0.000	0.204	0.356
KernelSVM+BoW	0.755	0.862	0.047	0.247	0.385
KernelSVM+Ling.	0.758	0.862	0.000	0.047	0.303
SentiWordnet	0.642	0.521	**0.127**	0.273	0.307
SentiStrength	0.664	0.564	0.106	**0.284**	0.318

the MVSA dataset. The reason is that statistical learning methods maximize the overall performance of classification, and thus may perform poorly on classes that are rare. This is consistent with the observation in [37]. In contrast, lexicon-based approaches are employed on each tweet independently. In some cases, it may be helpful to boost the performance of the rare class. Thus, the F-negative of SentiStrength is better than statistical learning approaches on MVSA, where negative instances are much less than other instances. Comparing the two kinds of features, we observe similar performance on the two datasets. Traditional BoW models the word distribution, which may ignore informative symbols in tweets. On the contrary, Linguistic feature explicitly represents the statistics on sentimental signals. However, some messages may not include the defined symbols. Hence a more advanced feature is needed for social data analysis. Finally, we observe that LinearSVM performs better than KernelSVM on tweet sentiment analysis. This is not surprising as the representations of short Twitter messages are very sparse. Mapping the feature into a high dimensional space results in a more sparse feature, which eventually hurts the performance. Since KernalSVM will incur much more computational cost, we adopt the efficient LinearSVM with BoW feature for sentiment analysis in our system.

4.5 Data visualization and user interaction

In our end-to-end system, we develop a web application to visualize the data and collect feedbacks from the users. The implemented functionalities are showed in Fig. 8. It can be easily extended to meet the requirements of other applications.

We collect the most up-to-date messages by using the Twitter streaming API, and perform sentiment analysis regularly. As social data is rich in content, an interface is designed to display the mined information in different dimensions. The development of the Web application follows the principle of Responsive Web Design (RWD) to provide an optimal view and interaction experience on different devices.

4.5.1 Visualization of aggregated information

As discussed in Section 4.4, the sentiment embedded in each message can be identified. Results from sentiment analysis is then aggregated with other information to form a complete picture on the topic. In this section, we present our system which aggregates information such as geolocation, timeline and topic.

Fig. 8 Applications implemented in the CDP

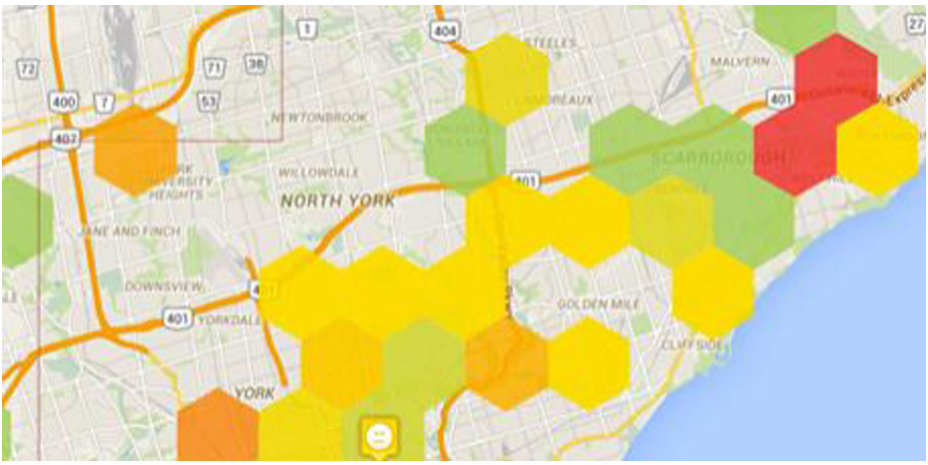


Fig. 9 The minimum area for visualizing the aggregated sentiment on the map is defined as a hexagon with fixed size. Larger area can be realized by further merging multiple hexagons

The minimum granularity of information aggregation with respect to the time and geographical area is set as day and a fixed sized hexagon respectively. Compared to a rectangle which is widely used for splitting the map, hexagonal grid can approximate the irregular boundary (e.g., city) more accurately. As the projection from 3D earth to 2D map will result in distortion, we define the hexagon directly on the plane coordinate of 2D map rather than the lat long coordinate. Figure 9 shows several defined hexagons on the map. In addition, we further index the hexagons in a hexagonal coordinate system, where each hexagon can be represented with three values corresponding to the three principle directions. The conversion between plane coordinate of 2D map and the hexagon coordinate can be easily computed.[7] Thus, we can quickly locate the hexagon of the coming Geo-tagged message using its plane coordinate provided by the GoogleMap API.

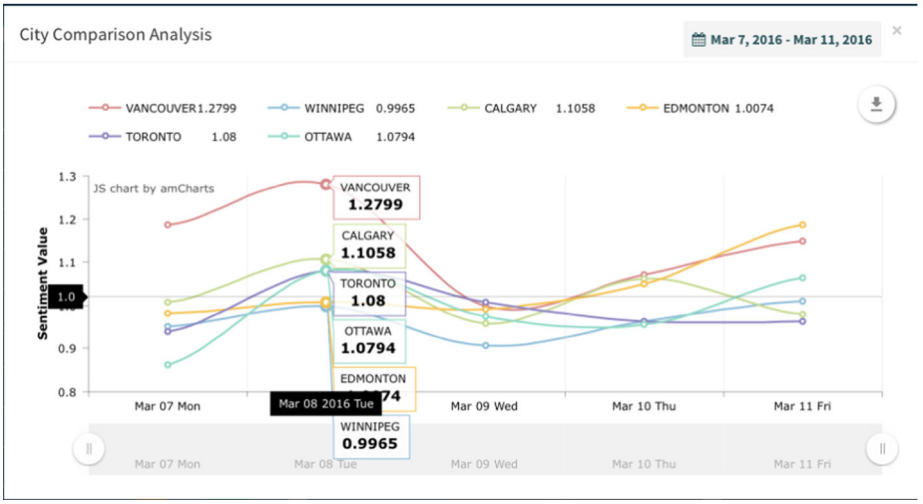
Figure 10 shows the overall sentiment of our selected 30 cities in Canada during a given period. In Fig. 10a, the cities are ranked based on their levels of positive sentiment. The number of messages from each city is also listed. Detailed daily information for selected cities can also be obtained as depicted in Fig. 10a. The trend of the public sentiments in multiple cities during the selected time period is showed in Fig. 10b. We can see that citizens in Vancouver are happier than other cities. There is a obvious peak for all the cities on March 08, which is the International Womens Day. With this dashboard, users can easily grasp the level of Web users' online activity and emotional status in different cities.

By clicking on a city in Fig. 10, we can get the city's detailed information as showed in Fig. 11. The top left image in Fig. 11a adopts a vivid change of colors to indicate changes in public sentiment in the selected city. The four images in Fig. 11b correspond to the sentiments detected in four days. The curve at the bottom of Fig. 11a indicates the overall sentiment changes during the time period. Each bar represents the number of messages, as well as the percentage of positive, neutral and negative sentiments. In addition, we also list the hot hashtags in the top right part of Fig. 11a. This can be used by users to understand the reason behind the sentiment change. Search results on Google and Twitter can be obtained

[7] http://www.redblobgames.com/grids/hexagons/.

Select	Rank	City	Score	Value	Select	Rank	City
	1	VANCOUVER	6039	1.2799		11	CAMBRIDGE
	2	BARRIE	869	1.2662		12	EDMONTON
	3	SURREY	1943	1.2375		13	WINNIPEG
	4	CALGARY	5207	1.1058		14	MONTREAL
	5	TORONTO	19862	1.0800		15	SAINT JOHN
	6	OTTAWA	5748	1.0794		16	WINDSOR
	7	HAMILTON	2375	1.0727		17	NIAGARA FALLS

(a) Rank of cities based their overall sentiment scores.



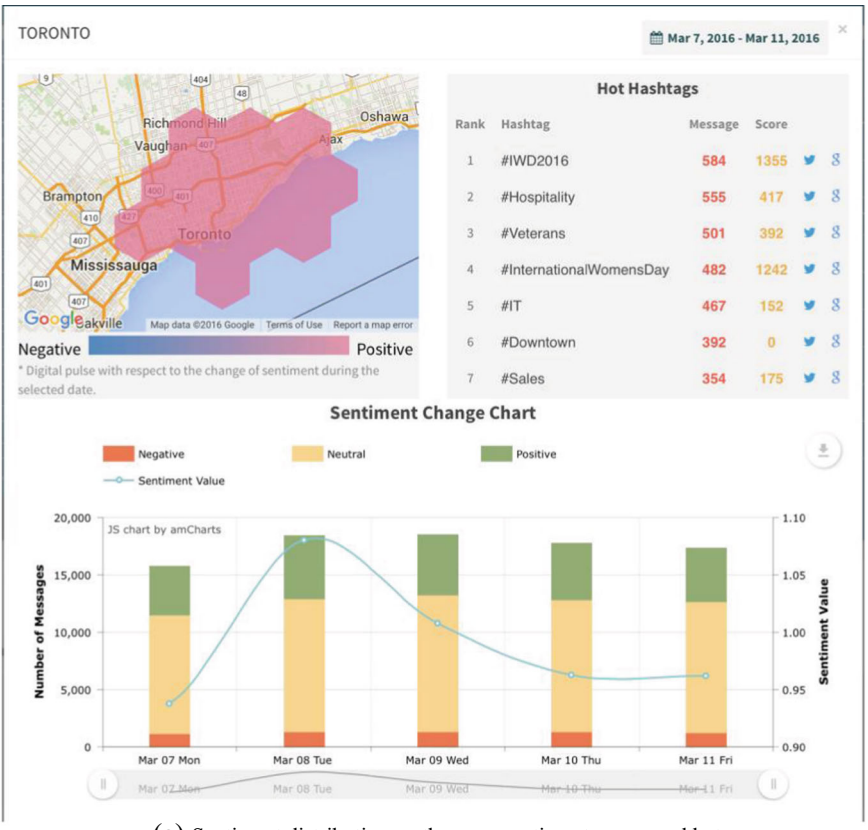
(b) Sentiment comparison of different cities during a given period.

Fig. 10 Comparison of sentiment in different cities during a selected time period

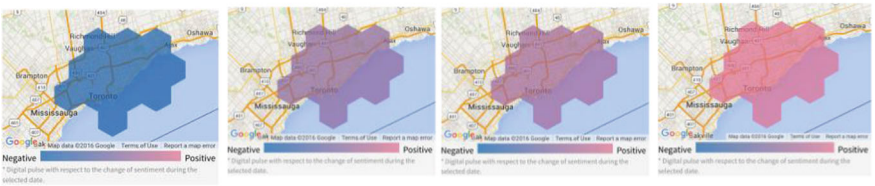
by clicking the icons after the hot hashtags. For example, the search results of "IWD2016" are showed in Fig. 11c and d respectively. By integrating the information from different platforms, the digital sentiment pulse and reasons behind the results of data analysis can be captured in a convenient way.

We can further narrow down the analysis by clicking the hashtag. The top left image in Fig. 12 will be updated to show the sentiment changes of messages containing the selected hashtag. Similarly, the curve and chart in the bottom image will also be updated accordingly. In addition, we list the images attached in the tweets of the selected topic in the top right part of Fig. 12.

In addition to the visualization of sentiment according to time, location and topic, we also provide other conditions, such as language and data source. In this way, the identified



(a) Sentiment distribution on the map, sentiment scores and hot hashtags for a given city.



(b) Dynamic visualization of detected sentiment during the given period

(c) Search result on Twitter for hot hashtag "IWD2016"

(d) Search result on Google for hot hashtag "IWD2016"

Fig. 11 The detailed information of each city

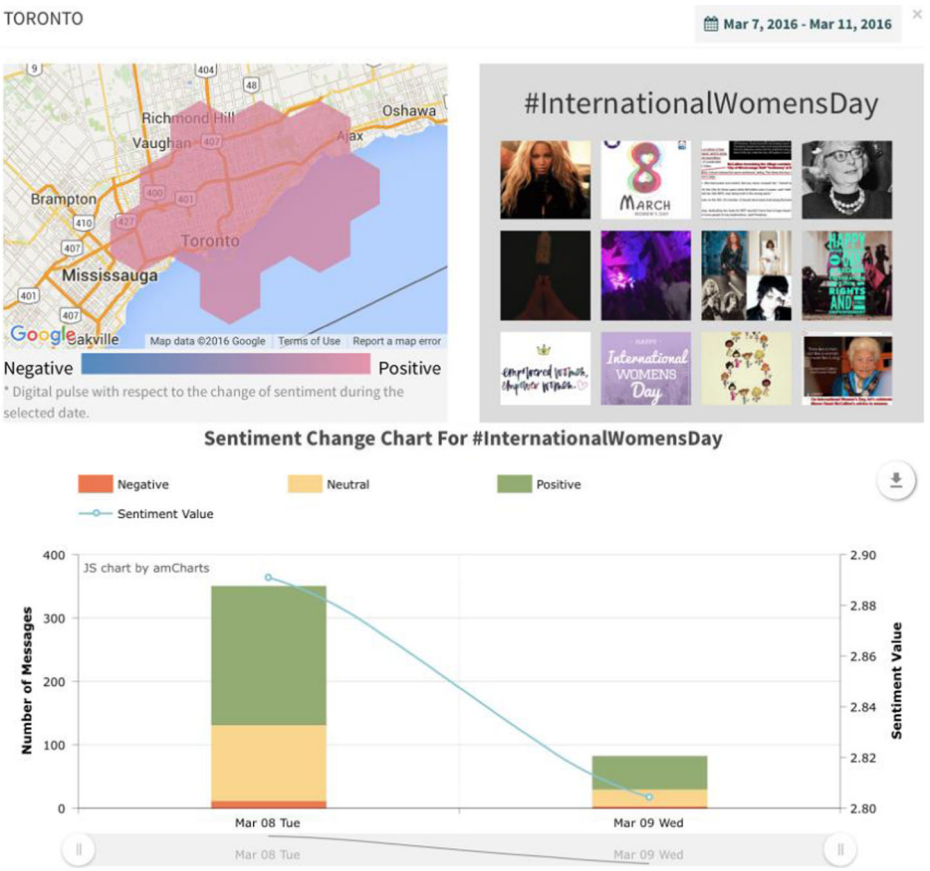


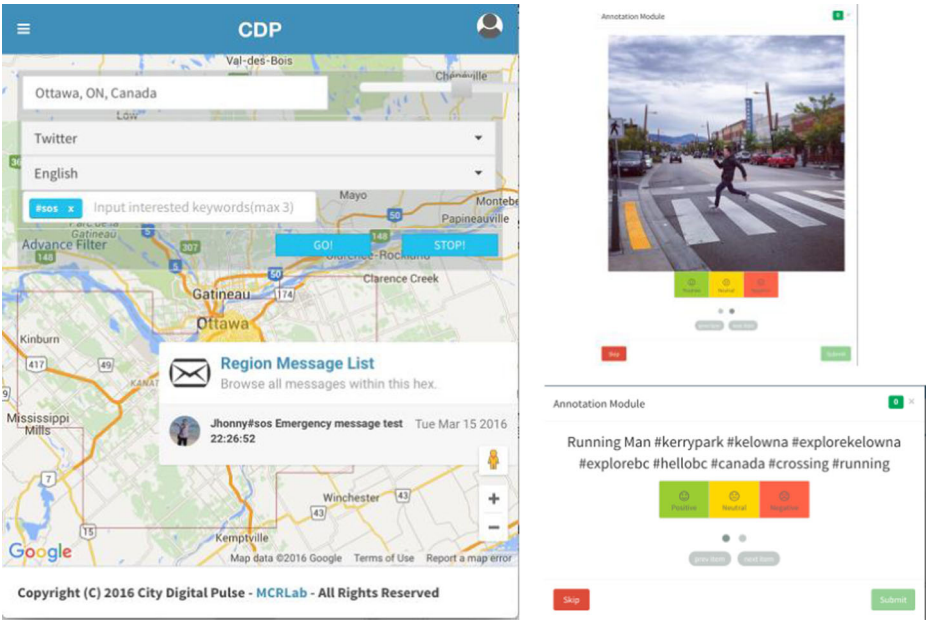
Fig. 12 The information corresponding to a selected hot hashtag

information can be presented to users in various dimensions and granularities. Different user requirements can be satisfied using our system.

4.5.2 Display of original messages

In this section, we introduce the functions of visualizing certain special messages, which can be emergency messages or messages involving national security. For example, Fig. 13a shows a message with the term "#SOS", which may be posted by citizens in danger. We can also monitor the social activities of pre-defined users (e.g., a known terrorist) on the map by tracking their posted messages.

To demonstrate the our system's capability on user interaction, we integrate a message annotation tool in our system. As showed in Fig. 13b, messages with accessible images are showed to Web users, who can contribute their annotations on the sentiment both on the textual and visual parts. The received annotations will be sent back to the database. Then the model for sentiment analysis will be re-trained and the sentiment scores of all messages will be updated using the new model. This training and prediction process will be performed regularly when enough new annotations are received.



(a) Emergency message visualization

(b) Annotation interface for filtered messages.

Fig. 13 Visualization of original messages and user interaction

4.6 System implementation and cloud based service

The overall system follows the front-end and back-end design principle. Front-end refers to the interface between users and the system. Back-end consists of the functional parts for responding the requests from Front-end, and accessing the data. Our interface is developed as Web applications. We adopt the single page application (SPA) framework, where front-end is developed by HTML and JavaScript, the communication between front-end and back-end follows the JSON RESTful protocol. Different from other frameworks, such as the Model-View-Controller (MVC), where HTML used in front-end is generated in the server side using PHP or JSP, SPA adopts a static webpage at the front-end. It only updates the parts affected by the new data, rather than the whole webpage. In this way, reloading the whole webpage is not necessary. For example, in Fig. 13a, only the new message will be communicated between front-end and back-end. Other parts such as the map and drop-down box will not be reloaded.

In addition to local deployment, we further deploy CDP on cloud server which can dynamically increase or decrease resources based on the system status. In our work, we select the Amazon Web Service (AWS),[8] which provides many services on various aspects of cloud-computing. For our system, the front-end employs the Amazon CloudFront service to distribute the content to end users with better performance. The back-end of our Web application is hosted on AWS Elastic Beanstalk. For our main SQL database, we adopt the the Amazon Relational Database Service for implementing the hard disk database. In

[8] https://aws.amazon.com/.

our system, we also design a cache database (memory database), which is deployed on the Amazon ElastiCache service. As commercial cloud service is expensive, our current system on the cloud only keeps the timely statistical information explored from the collected data. The original data is kept on the local server. All the functions discussed in Section 4.5 are implemented on both the local server[9] and Amazon cloud.[10]

5 Conclusion and future work

In this paper, we have proposed our City Digital Pulse (CDP) platform. As an end-to-end system offering various applications for a smart city, the design follows several principles which are essential for addressing the challenges raised by the large-scale heterogeneous data generated by different sensors in a smart city. The system covers the whole data science process, ranging from data collection to data analysis. As proof of concept, we deploy platform to analyze social data. A web application is then developed to monitor citizens' sentiments from posted social messages.The analysis on the large pools of tweets resulted is then merged with other supporting information such as time and geolocation. The multi-faceted information is then visualized in an easy-to-digest manner for user consumption. Our system has been deployed on both local server and Amazon cloud which are accessible by the public.

We have only considered soft sensors (Twitter and Instagram) in our current work. Future extension includes the integration of hard sensors such as the smart object in the smart city. More innovative applications can also be developed on our platform. Future research topics include fusion methods on the various contextually related data, and the system load balance. Another important issue to study is how to perform computationally intensive cloud-computing techniques for data analysis.

References

1. Agrawal R, Kadadi A, Dai X, Andres F (2015) Challenges and opportunities with big data visualization. In: Proceedings of the 7th International Conference on Management of computational and collective intelligence in digital ecosystems. ACM, pp 169–173
2. Borth D, Ji R, Chen T, Breuel T, Chang SF (2013) Large-scale visual sentiment ontology and detectors using adjective noun pairs. In: ACM MM
3. Buzzi M, Buzzi M, Franchi D, Gazzè D., Iervasi G, Marchetti A, Pingitore A, Tesconi M (2014) Big data: a survey. Mobile Netw Appl 19(2):171–209
4. Buzzi M, Buzzi M, Franchi D, Gazzè D., Iervasi G, Marchetti A, Pingitore A, Tesconi M (2016) Facebook: a new tool for collecting health data? Multimedia Tools Appl 1–24
5. Castro1 M, Jara1 AJ, Skarmeta AFG (2013) Smart lighting solutions for smart cities. In: International conference on advanced information networking and applications workshops
6. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: from big data to big impact. MIS Q 36(4):1165–1188
7. Chen T, Lu D, Kan MY, Cui P (2013) Understanding and classifying image tweets. In: ACM MM
8. Chen T, SalahEldeen HM, He X, Kan MY, Lu D (2015) VELDA: relating an image tweet's text and images. In: AAAI
9. Costa C, Santos MY (2015) Improving cities sustainability through the use of data mining in a context of big city data. In: Proceedings of the world congress on engineering
10. Dave K, Lawrence S, Pennock DM (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW

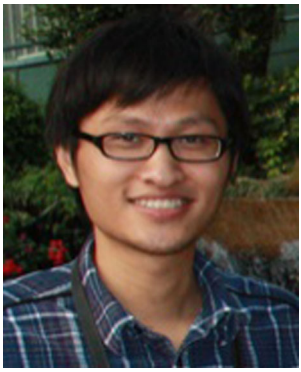
[9] http://citypulse1.site.uottawa.ca.

[10] http://http://citydigitalpulse.us-west-2.elasticbeanstalk.com.

11. Dey S, Chakraborty A, Naskar S, Misra P (2012) Smart city surveillance: leveraging benefits of cloud data stores. In: IEEE 37th conference on local computer networks workshops (LCN Workshops)
12. Fan M, Sun J, Zhou B, Chen M (2016) The smart health initiative in china: the case of wuhan, hubei province. J Med Syst 40(3):62:1–62:17
13. Fang X, Zhan J (2015) Sentiment analysis using product review data. J Big Data 1–14
14. Fang Q, Sang J, Xu C, Hossain MS (2015) Relational user attribute inference in social media. IEEE Trans Multimedia 17(7):1031–1044
15. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. Processing 1–6
16. Hossain MS, Muhammad G, Al Hamid MF, Song B (2016) Audio-visual emotion recognition using big data towards 5G. Mobile Networks and Applications
17. Hossain MS, Muhammad G, Song B, Hassan MM, Alelaiwi A, Almari A (2015) Audio-visual emotion-aware cloud gaming framework. IEEE Trans Circuits Syst Video Technol 25(12):2105–2118
18. Hromic H, Phuoc DL, Serrano M, Antonic A, Zarko IP, Hayes C, Decker S (2015) Real time analysis of sensor data for the internet of things by means of clustering and event processing. In: ICC
19. Hsu CY, Yang CS, Yu LC, Lin CF, Yao HH, Chen DY, Lai KR, Chang PC (2015) Development of a cloud-based service framework for energy conservation in a sustainable intelligent transportation system. Int J Prod Econ 164:454–461
20. Hu X, Tang L, Tang J, Liu H (2013) Exploiting social relations for sentiment analysis in microblogging. In: WSDM
21. Hwang D, Jung JE, Park S, Nguyen HT (2015) Social data visualization system for understanding diffusion patterns on twitter: a case study on korean enterprises. Comput Inf 33(3):591–608
22. Jiang Y, Xu B, Xue X (2014) Predicting emotions in user-generated videos. In: AAAI
23. Khan Z, Anjum A, Kiani SL (2013) Cloud based big data analytics for smart future cities. In: International conference on utility and cloud computing
24. Lê Tu'n A, Quoc HNM, Serrano M, Hauswirth M, Soldatos J, Papaioannou T, Aberer K (2012) Global sensor modeling and constrained application methods enabling cloud-based open space smart services. In: 9th international conference on ubiquitous intelligence & computing and 9th international conference on autonomic & trusted computing (UIC/ATC), 2012, pp 196–203
25. Lombardia P, Giordanob S, Farouhc H, Yousefd W (2012) Modelling the smart city performance. Innov Eur J Soc Sci Res 25(2):137–149
26. Ma S, Liang Z (2015) Design and implementation of smart city big data processing platform based on distributed architecture. In: International conference on intelligent systems and knowledge engineering
27. Mell PM, Grance T (2011) Sp 800-145. the nist definition of cloud computing. Tech. rep., Gaithersburg, MD, United States
28. Mukkamala RR, Sørensen JI, Hussain A, Vatrapu R (2015) Detecting corporate social media crises on facebook using social set analysis. In: 2015 IEEE international congress on big data. IEEE, pp 745–748
29. Mullen T, Collier N (2004) Sentiment analysis using support vector machines with diverse information sources. In: EMNLP
30. Niu T, Zhu S, Pang L, El-Saddik A (2016) Sentiment analysis on multi-view social data. In: MultiMedia modeling, pp 15–27
31. Nuaimi EA, Neyadi HA, Mohamed N, Al-Jaroodi J (2015) Applications of big data to smart cities. J Internet Serv Appl 6–25
32. Palmieri F, Ficco M, Pardi S, Castiglione A (2016) A cloud-based architecture for emergency management and first responders localization in smart city environments. Comput Electr Eng
33. Pang B, Lee L (2007) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
34. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: EMNLP
35. Plotnikova N, Kohl M, Volkert K, Lerner A, Dykes N, Ermer H, Evert S (2015) KLUEless: polarity classification and association. SemEval 2015 workshop
36. Rosenthal S, Nakov P, Kiritchenko S, Mohammad SM, Ritter A, Stoyanov V (2015) SemEval-2015 task 10: sentiment analysis in twitter. SemEval 2015 workshop
37. Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. ESSEM workshop
38. Saini M, Alam KM, Guo H, Alelaiwi A, Saddik AE (2016) Incloud: a cloud-based middleware for vehicular infotainment systems. Multimedia Tools Appl 1–29
39. Scholl HJ, AlAwadhi S (2016) Smart governance as key to multi-jurisdictional smart city initiatives: The case of the ecitygov alliance. Soc Sci Inf 55(2):255–277
40. Su K, Li J, Fu H (2011) Smart city and the applications. In: ICECC

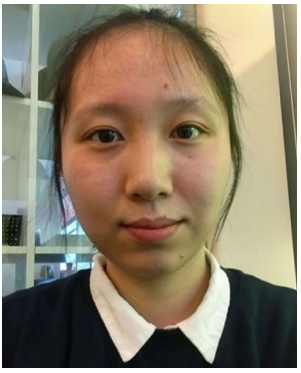
41. Sudhof M, Goméz Emilsson A, Maas AL, Potts C (2014) Sentiment expression conditioned by affective transitions and social forces. In: SIGKDD
42. Taherkordi A, Eliassen F (2016) Scalable modeling of cloud-based iot services for smart cities. In: 2016 IEEE international conference on pervasive computing and communication workshops, percom workshops 2016, pp 1–6
43. Tedeschi A, Benedetto F (2014) A cloud-based tool for brand monitoring in social networks. In: International conference on future internet of things and cloud (FiCloud), 2014, pp 541–546
44. Wen Y, Zhu X, Rodrigues JJPC, Chen CW (2014) Cloud mobile media: reflections and outlook. IEEE Trans Multimedia 16(4):885–902
45. Yamamoto S, Nakamura M, Matsumoto S (2012) Using cloud technologies for large-scale house data in smart city. In: International conference on cloud computing technology and science (CloudCom)
46. Yang J, He S, Lin Y, Lv Z (2015) Multimedia cloud transmission and storage system based on internet of things. Multimedia Tools Appl 1–16
47. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M (2014) Internet of things for smart cities. IEEE Internet Things J 1(1):22–32
48. Zhao Y, Qin B, Liu T, Tang D (2014) Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog. Multimedia Tools Appl 1–18

Zhongli Li received his B.Eng. in Computer Engineering from Central South University in 2013 and started his M.Sc in University of Ottawa at the same year. He is currently working at Multimedia Computing Research Laboratories (MCRLab), University of Ottawa, supervised by Professor Abdulmotaleb El Saddik. His research interests include Iot, big data, and machine learning.

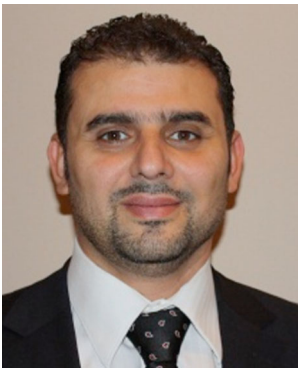


Shiai Zhu is currently a Post-doc Researcher in the Multimedia Computing Research Laboratories (MCRLab) at University of Ottawa. He received his PhD degree in Computer Science from the City University of Hong Kong, M.S. degree from University of Science and Technology of China and B.S. degree from Civil Aviation University of China. Before joining University of Ottawa, he was a Staff Researcher at Image & Visual Computing Lab (IVCL) Hong Kong, Lenovo Group. His research interests include social media, multimedia analysis and machine learning. Specifically, he focuses on research of image and video content understanding and its applications.

Huiwen Hong received her B.Eng. in Computer Science from Dalian University of Technology in 2014 and started her Master of Computer Science (MCS) in University of Ottawa at the same year. She is currently working at Multimedia Computing Research Laboratories (MCRLab), University of Ottawa, supervised by Professor Abdulmotaleb El Saddik. Her works include front-end web development and user experiences design.



Yuanyuan Li, received the B.E. degree from Southwest Jiaotong University, Sichuan, China in 2014 and started study for her M.Sc degree at University of Ottawa at the same year. She's currently working at the Multimedia Computing Research Laboratories (MCRLab), University of Ottawa, supervised by Professor Abdulmotaleb El Saddik. Her main areas of research interest are big data and data base.



Abdulmotaleb El Saddik, (F'2009) is University Research Chair and Distinguished Professor in the School of Electrical Engineering and Computer Science at the University of Ottawa. He is an internationally-recognized scholar who has made strong contributions to the knowledge and understanding of multimedia computing, communications and applications. He has authored and coauthored four books and more than 450 publications. Chaired more than 40 conferences and workshop and has received research grants and contracts totaling more than $18 Mio. He has supervised more than 100 researchers. He received several international awards among others ACM Distinguished Scientist, Fellow of the Engineering Institute of Canada, and Fellow of the Canadian Academy of Engineers and Fellow of IEEE and IEEE Canada Computer Medal.