NaviSoc: A Socially Enhanced Real-time Navigator

Nikolaos Louloudakis*†, Vasileios Theodosiadis*†, Haridimos Kondylakis†, and Kostas Stefanidis†
*Computer Science Department, University of Crete, Greece
Email: {luludak, vtheodos}@csd.uoc.gr
†Institute of Computer Science, FORTH, Greece
Email: {kondylak, kstef}@ics.forth.gr

***Abstract*—As the usage of social networks becomes more and more ubiquitous and people commute more often today, social streams have become a valuable source for many kinds of applications. For example, the various social streams could be exploited for choosing the optimal path (e.g., the shortest and/or the fastest) to reach a desired destination. To this direction, we present a novel navigation system, called NaviSoc. NaviSoc is a location-based, context-aware recommendation system proposing dynamically and in real-time, the best route according to the user's context (e.g., preferences and budget). The system takes advantage of social streams in order to identify unprecedented (e.g., traffic jam, protest) or social events occurring in a specific place. Then, using this knowledge, the system dynamically readjusts the proposed navigation path. An initial evaluation, performed on the Greek island of Crete, demonstrates the feasibility of our solution and the benefits of our system.**

I. Introduction

Nowadays, Global Navigation Satellite Systems (GNSS) allow users to worldwide pinpoint their locations or the locations of other people and goods at any given moment. Those systems are used more and more in order to relieve traffic conditions, improve the efficiency of vehicle use, speed up rescue operations for people in distress [19], and provide tools for meteorology [8], geodesy, and many others. However, although GNSS systems recognize the importance of context in navigation (e.g., user preferences) and real-time event identification (e.g., strikes) the approaches already implemented are inefficient with little or no impact. For example, the widely used vehicle GPS routing systems work in a more simplified, static, and less accurate manner without considering real-time events and user's context.

Recently however, with the digitization of communications and social relationships, people use more and more smart devices and mobile phones, for social networking, navigation and gaming. The proliferation of mobile devices leads to an application-driven society, with an average smartphone user having installed and using about 26 applications per month [6], presenting marketers with new opportunities to collect user preferences and habits. This proliferation of mobile devices has raised the interest of store owners and companies to become advertized and promoted via mobile applications in different ways, such as having developed their trademark applications or placing advertisements on existing mobile applications. On the other hand, new research tools facilitate the extraction and interpretation of information published from those devices on social streams. There is already a mass amount of research solutions on event identification using such streams, reporting that the burstiness of these streams and the psychosynthesis of users when writing those streams may denote an event taking place at their location [16]. Twitter, for example, has been viewed as a social sensor detecting catastrophic events [20], predicting the stock market [7], and detecting real-time events [17].

Despite the large amount of work on these areas, to our knowledge, there is no other implemented system allowing the combination of user context and social streams in order to provide real time navigation. Currently running European projects [10], [9] focus on similar goals to NaviSoc, showing the emerging demand for the combination of such technologies. NaviSoc contributes to both social and business welfare as it aims to suggest dynamic routes to commuters based on their context and current circumstances by taking advantage of social networks. Simultaneously, shop owners can benefit from NaviSoc as the system can also be used for promotion purposes.

In a nutshell, NaviSoc is an innovative, socially-enhanced, real-time navigation system that tackles the aforementioned challenges effectively and efficiently. More specifically:

- We demonstrate a system able to estimate the time a commuter needs to arrive at her destination and propose the best possible route depending on the commuter's context and the eventual circumstances of each potential route. The commuter needs just a smartphone with the client application of NaviSoc and access to the Internet, even via a low-bandwidth connection.
- The proposed system takes into account the user's context (e.g., location, interests, and willingness-to-pay) as well as events in potential routes that the user might want to attend in order to go from a place to another. By combining them, the system can suggest the most advantageous route for the user in terms of time traveling, safety, and user interests.
- To detect events, we used state of the art machine learning algorithms (Support Vector Machine, Latent Dirichlet Allocation) on social streams that are stored in a central database.
- The central database uses NoSQL technologies to handle the enormous dataflow of social streams. According to tweespeed.com, there are now more than 15,000 tweets per minute published in Twitter. Social streams are pooled continuously and the data are stored to our local database. For the time being Twitter is used but there is no restriction on the specific social stream used.

- Finally, to determine user context, the description and the category of the applications, already stored in user's smartphone are accessed and used.

The system is based on the existing widespread usage of social media (e.g., Twitter), state-of-the-art wireless/satellite equipment (e.g., IEEE 802.11 access points, GNSS, and cellular towers) and uses machine learning algorithms to detect and identify real-time events, taking into consideration two facts: a) that people are commuting more and more often and b) that the social network streams provide big data flows with useful information.

The remaining of this paper is structured as follows: In Section 2, we present a motivation example, while in Section 3, we describe the system architecture and the building blocks of our platform. In Section 4, we demonstrate the system implementation and present some initial evaluation results. Section 5 presents the related work and Section 6 provides a summary of our work and gives an outlook for further work.

II. Motivating Example

For the sake of simplicity, and without loss of generality, we describe our paradigm using the case of wandering in Crete, Greece, a place visited by many tourists with different preferences (e.g., interests, budget). Clearly, our solution is easily applicable in metropolitan areas as well, especially the ones characterized mostly by congestion or social events. In fact, we strongly theorize that moving in the cities of the island of Crete can precisely describe our solution to a common problem.

Crete is the biggest island of Greece, a Greek administrative region, and one of the most popular tourist destinations in Europe. Millions of tourists visit it annually, according to the Hellenic National Statistics Agency. Some of them are interested in visiting this island because of the existence of some events every year (e.g., Choudetsi festival). Due to the majority of activities that both tourists and locals can occupy, it is critical to discriminate interesting ones for users of NaviSoc, based on their context. For example, a user might be located near the capital city of Crete, Heraklion, she is not willing to visit places located far away and she is also interested in greek music. Thus, NaviSoc will recommend relative concerts as well as traditional festivals. For simplicity, we consider here that the user can be served only by public transportation (e.g., urban and intercity buses).

NaviSoc detects events via users' tweets and context. Twitter is constantly used by people that usually post tweets when an event takes place within a small amount of time. The type of event could vary. In our case, we consider events which affect commuting, such as protests, accidents, and strikes and social events, such as festivals. For the sake of brevity, we speculate that due to force majeure a concert in Pyli Bethlehem Theater just cancelled. NaviSoc detects this unprecedented situation and proposes another place, e.g., Technopolis Hall, with relative concerts being accessible by the public local transportation. As traveling is sometimes time-consuming, people sleep, gaze or even not continuously search for their final destination. The NaviSoc's functionality that warns the user some minutes after the event is detected and before she reaches her destination, could be very useful to remedy this problem.

In addition, NaviSoc also contributes to the business welfare. It can promote companies, which may want to be advertised (e.g., restaurants, hotels) in the nearby area. Thus, a user that is interested in the services of these companies can be easily informed.

III. The NaviSoc System Overview

NaviSoc focuses on recommending the best possible route for a user based on the user's context and the events that are identified from meaningful information retrieved using social streams. In this section, we present the design of the main components of our application. NaviSoc is based on a server-client architecture, as shown in Fig. 1. Concerning the server side, it consists of the following components: the Social Stream Retriever, the Storage Handler, the Event Detector, the GeoPlanet Retriever, the Users Content and Context Storage Handler, and the Server Communication Handler. On the other hand, the Client consists of the Graphical User Interface (GUI), for user navigation, and the Back-End component, for data processing and communication with the server. Next, we will describe and analyze all these components in detail.

A. *The Social Stream Retriever*

The Social Stream Retriever is responsible for stream sampling from well-known social applications, such as Twitter and Facebook. In our proof-of-concept prototype, we used the Twitter REST API to collect tweets from selected social streams, taking into consideration their provision of valid and timely information for a specific area (in our case, the island of Crete). Thus, false positive rate can be reduced detecting yet many events taking place in the area. In fact, this component converts the tweets retrieved in JSON format from the streams into a format that can be easily saved in NaviSoc's social streams database system by using a relative small wrapper component and pass the formatted data to the Storage Handler component. For sake of simplicity, we can consider a tweet having the content "Concerts in Pyli Bethlehem Theater have canceled due to force majeure". The component will retrieve this tweet from the social stream and give it as input to the Tweets Storage Handler. To lower our susceptibility to rumors and bad information, it is possible to identify and register trusted accounts that will be periodically checked for new information. As tweets are characterized by decreased quality of data, and humans are usually not as reliable as other sensors, we diversify the source selection, as proposed in [21], in order to suppress more rumors. Apart from that, the tweets of these sources are not characterized by misspelling and abbreviations that may pose an extra challenge to the event detection process.

In order to maximize the effectiveness of the sampling, avoiding the retrieval of already retrieved content, we consider taking small content samples periodically. The accounts that we used in our example (Texnopolis, Cretevents, Rethemnos, CRETAZINE, cityofheraklion) are social accounts of high trust, providing timely information about social, organized events or even real-time unexpected events happening in the broader area of the island of Crete. Our sampling process proceeds as follows: the system samples the social accounts

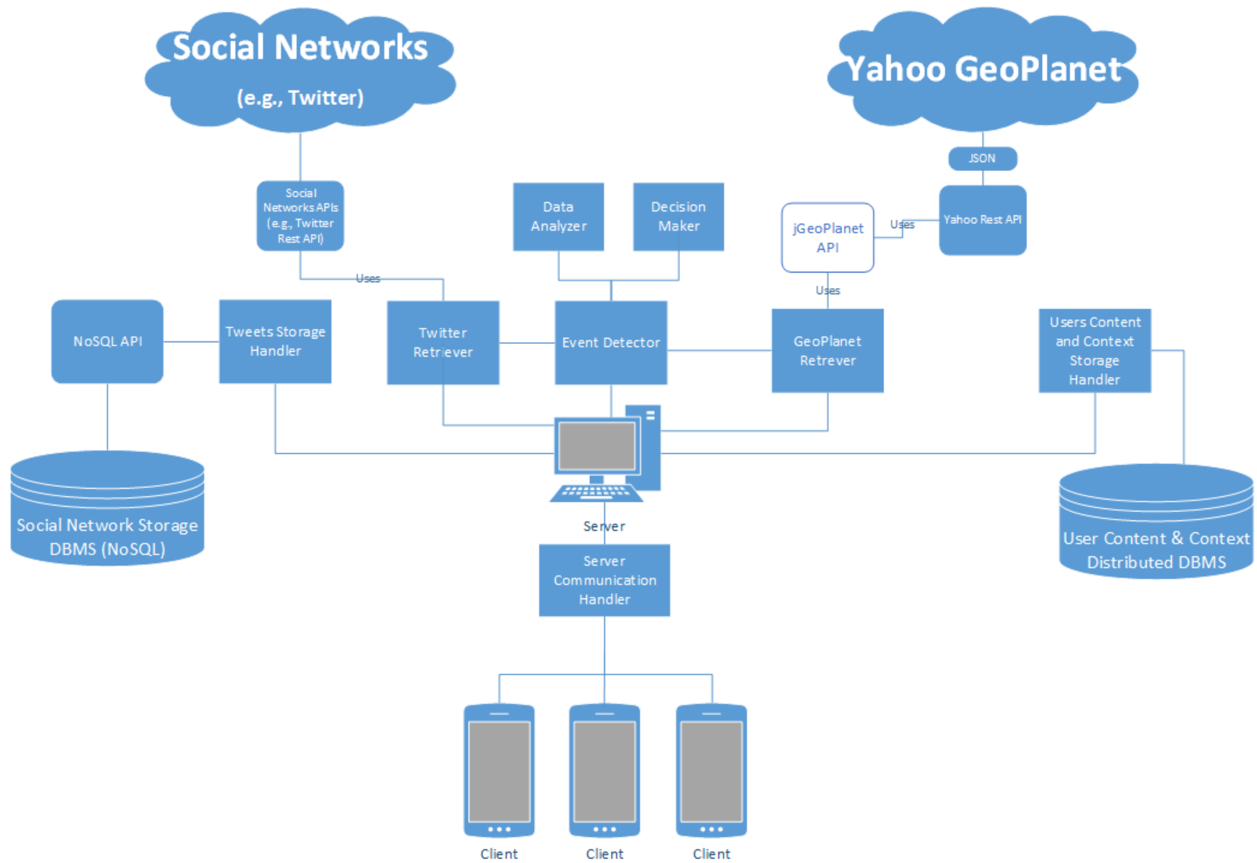

Fig. 1. The NaviSoc System Architecture.

for new tweets, along with relevant information, such as the unique id of the tweet in the social network, its date created, the number of its hashtags and the number of its retweets, every 15 minutes. In between, the system turns into an inactive status, minimizing the usage of the server resources. The number of tweets requested in each sampling procedure is 20 per source. Both polling time and sampling size depend on the average rate of new tweets and are dynamically adapted by our system in order to avoid potential content loss in the sources between the sampling procedures, in case of dense tweet posting. On the other hand, the system detects and discards tweets that are already retrieved in a previous sampling procedure, avoiding duplicates.

B. The Storage Handler

The Storage Handler manages the storage process of the data downloaded by the Social Stream Retriever in the NaviSoc's database (Social Network Storage NoSQL DBMS) using a NoSQL API. The Handler consists of a distributed, Not-Only SQL Database Management System responsible for the storage of the retrieved tweets profiting from saving the data in the same format retrieved without integration in a distributed manner, preserving the data Atomicity, Consistency, Isolation, and Durability standards. This information is accessed by machine learning and decision making algorithms in order an event to be detected and finally recommended to the user client application. In this stage, words or phrases from the specific tweet will be saved in particular fields of the database. For instance, if the word "Pyli Bethlehem" is detected in a tweet, this value will be applied to the GeoPlanet component, in which, in combination with a location recognition algorithm, its coordinates will be detected and stored in the database in specific fields for coordinates. This information can be used, in combination with other words in the tweet, such as "concert", to trigger an event detection in the Event Detector component of the system.

C. The GeoPlanet Retriever

The GeoPlanet Retriever uses the homonym Web service of Yahoo and is responsible to search for locations (coordinates, region, country) by taking as input a specific word - candidate location and using the jgeoplanet API provided by Winterwell [3]. In other words, it contributes to location detection by taking advantage of words/phrases. Thus, this component is used to retrieve the location of events, so that they can be pinpointed on a map. As mentioned before, a word submitted to the component (e.g., "Pyli Bethlehem") will be analyzed in order to retrieve also potential variations of the word that can correspond to a location. Then, the word and its variations will be submitted to the Yahoo GeoPlanet service which will attempt to identify the specific location. If none of the words and their derivatives corresponds to a location, they will be discarded. Our future plans include extending the system to detect events in more sophisticated and efficient ways (e.g.

stripping out the articles) by taking also into consideration more than two words.

D. The Event Detector

The Event Detector component is responsible for the event detection process. An event can be associated with either a social festivity or an unprecedented situation, which may be an obstacle to the scheduled route of a user (e.g., protest, construction work). To be more easily perceptible, the Data Analyzer and the Decision Maker components can be used either in combination or independently. The first component is responsible for the semantic analysis while the second is in charge of applying out machine learning algorithms.

Semantic analysis targets to identify entities related to categories of our interest (e.g., means of public transportation, concert halls) and uses the X-Link [12] web service, a fully configurable, linked data-based, named entity extraction tool. Thus, we group both the entities of interest and the ones without any value for our system, in order to enrich our training local database.

Then the Decision Maker component uses the training local database to apply machine learning algorithms (e.g., Support Vector Machines) in order to detect potential events. Our algorithms take into consideration the words of a tweet and their combinations, but also the events already detected in other tweets. For instance, the words "Pyli Bethlehem" and "concert" might trigger an event about a concert in the Pyli Bethlehem place in the system, especially if this event is potentially "detected" from other tweets.

E. The Users Content and Context Handler

The Users Content and Context Handler is responsible for the server-side database storage of the user's information about context (e.g., interests, preferences, location), contributing to recommendations produced by NaviSoc. This component obtains the information given from the clients via the Client Communication Handler component and saves it in a distributed DBMS system in a specified form.

F. The Server Communication Handler

The Server Communication Handler is a component responsible for the communication management of the server of the system with its clients. This component receives all the content and context information from those clients and forwards it to the appropriate components, in order to be stored in the appropriate database management system so that they can be used from the system to personalize the event detection process. The component is also responsible for updating the client applications states (detected events to be shown in the application) and data (context additions in order to be stored in the local client database). Additionally, the component takes the role of the management of the clients registering and unregistering to the server, so that the server knows all the clients able to communicate with it.

G. The Client Application

On the client side, the application consists of the GUI and the back-end component (Fig. 2). The GUI is responsible for displaying the relevant information about events to the user. The back-end component is responsible for a) retrieving the user's context information, b) storing this information at a local lightweight database, and c) communicating with the server in order to share this information.

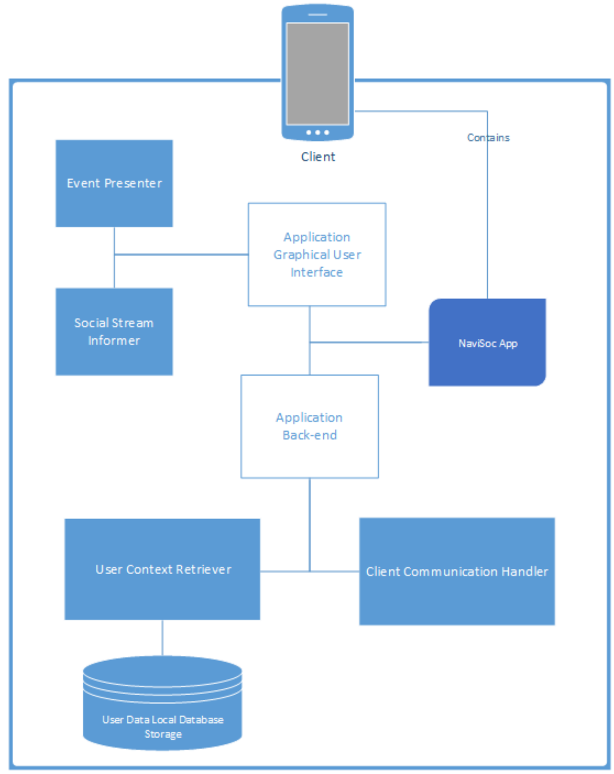

Fig. 2. The Client Application of the NaviSoc System.

1) The GUI: The GUI consists of the Event Presenter and the Social Stream Informer. The Event Presenter is responsible for displaying the detected events from the system to the user on a map, based on her location and with discrete signs and markers. Apart from those signs, additional information is provided to the user, such as the specific social stream elements used to detect the individual events and the time detected. This is accomplished using the Social Stream Informer, a component that presents a number of fresh Social Stream elements retrieved from the trusted accounts on the server, giving to the user the opportunity to get informed by reading them textually and not on the map.

2) The Back-end: The back-end consists of the User Context Retriever and the Client Communication Handler. The User Context Retriever obtains the user context, such as location, budget and interests from mobile state and sensors, like GPS. To identify the interests and the context of the user the installed applications descriptions and categories are identified and used. Moreover, to identify the location of the user, GNSS (e.g., GPS, GLONASS) measurements, offered by most mobile devices are employed. However, the modular design of our client allows advanced localization techniques (indoor for metro, outdoor otherwise) and triangulation to be also integrated in the future. For example, Received Signal Strength Indication (RSSI) fingerprint based on confidence intervals,

Fig. 3. A user interface example of the NaviSoc System.

percentiles, empirical distribution of RSSI measurements, and theoretical distributions, such as Multivariate Gaussian, can be used. All information detected is stored in the User Data Local Database Storage, using an SQLite database. The Client Communication Handler communicates with the server in order to send the user content and context information to the Server Communication component. The component requests and retrieves from the same server component all the event information that will be managed and finally displayed in the application front-end.

IV. Implementation and Evaluation

As already described, NaviSoc is based on a server-client architecture. The server was implemented using Java v7, whereas the web application part for the clients was implemented using PHP, HTML with CSS, and JavaScript. The Google Maps API was also used for drawing events and paths on Google Maps.

Using our application a user is able to get informed about events happening near her area, giving her the opportunity to make event attending plans and actions, such as buying tickets, on time. In addition, our application can be used as a real time navigator, showing also nearby points of interest according to the user's context. Whenever an unexpected event takes place the route is rescheduled and alternative navigation paths are proposed. As a simple example consider the application interface shown in Fig. 3.

To evaluate our system, we used a workstation running Windows 7 with an Intel Pentium, 4th generation dual-core processor at 3.2 GHz, 8GB of memory, and 2 SATA II Hard Disk Drives of 5200 RPM. The workstation had installed a WAMP application server platform, containing Apache v.2.4.9 and PHP v5.5.12 and the NoSQL database used is Apache Cassandra 2.0.9.

In order to collect the tweets needed to detect events dynamically, we executed the following steps. First of all, the system gets into an active status phase every 15 minutes. While being in that phase the system retrieves a number of tweets (100, in total) from each one of the stream sources sources - the number of retrieved tweets is dynamically adapted according to the flow of the tweets. After that, the system converts the Greek characters of each tweet into Latin, using a small UTF-8 conversion component, and checks the converted tweet content for interesting events. If no event is found in the tweet, the tweet is stored in the database, being marked as of no interest. Otherwise, we check if the keyword found in the tweet corresponds to a statically defined location. Statically defined locations are a set of locations manually inserted into the database in order to increase the location detection of a potential event, as the GeoPlanet service lacks on precision concerning points of interest or city streets. When a location is identified, it is stored as the location of the tweet. Otherwise, our system tries to determine whether or not any of the words of the tweet corresponds to a location, by splitting the converted tweet into words, filtering them by size for efficiency and submitting each word to the GeoPlanet service. If a location is identified by GeoPlanet, that location is stored into the database as the detected location of the tweet as well. To provide information to the end-users, user context and location is identified through the client application and a script dynamically retrieves nearby detected events. All information retrieved is visualized on a map that is provided to the user.

As a proof of concept experiment, we collected 304 tweets from specific stream sources of high trustworthiness (e.g., Cretevents), collected over 4 days. We tried to limit the number of tweets processed in order to be able to manually identify as well the correctness of our method. Concerning the frequency of retrieval, it was set to 15 minutes, applied periodically. As already stated this number is adapted periodically according to the flow of the tweets as it is of high importance to detect fresh events on time. As a result, we scanned 304 tweets, 147 of them containing at least a keyword of interest, and 93 of which were located in the region we are interested in, resulting to an event detection. From the detected events, a percentage of 91% was correctly detected. The rest of them mainly referred to advertisements. We observed that the majority of the events were located near the city of Heraklion, as they are the 73.1 % of the total detected tweets in Crete. Concerning the other prefectures of the island, 19, 4 and 0 events detected in Rethymno, Chania and Lasithi, respectively. Not surprisingly, the majority of tweets were located in Heraklion, which is the largest city and the capital of the administrative region. This is reasonable as well because, besides large cities, Crete consists of small villages with low social media and technology penetration.

Most of the events, which are not detected by NaviSoc, are not related to social or unprecedented events since they refer to news of Crete or generally Greece, related to political and of development content. Only a small percentage is missed by our algorithm including tweets containing only hyperlinks to other sites. Moreover, due to the restriction of the size of tweets (twitter allows only 140 characters per tweet), there were also cases using only a part of the word related to a location. As a consequence, NaviSoc could not detect the related events.

V. Related Work

Event detection from social streams has been the topic of extensive research. Li et al. [16] focused on a mechanism that detects events from the social stream of Twitter by analyzing and clustering tweets. In addition, social streams (e.g., Twitter) have been exploited to predict the stock market [7], to detect events about catastrophic phenomena [20] and to protect the inhabitants from being affected from these disastrous phenomena, as this mechanism warns them ahead of time. In fact, Toretter [20] can inform inhabitants of a radius of 100 km away of an earthquake epicentre ahead of time – earlier than Japan Meteorological Authority (JMA). However, NaviSoc is not only targeted for event detection, but uses the detected events in accordance to the user's context to dynamically present navigation information to the user.

Besides event detection, dynamic routing and localization is also the target of several applications. For example, in terms of indoor localization, Travi-Navi [1] "learns" the area from sensor readings, records of high-quality images, and packs all these images in a navigation trace without taking into consideration indoor localisation system and floor maps. This is useful for supermarkets and big stores. In addition, there are some outdoor recommendation systems for dynamic routing. Waze [14], for instance, is based on OpenStreetMap (OSM) [5], using the Kort application [4] to populate its database with reports of road hazards and collaborative management of vehicle flow. The Kort application uses the concept of gamification to motivate users to use it (users are rewarded by coins when reporting unprecedented events) and humanitarian volunteers are also able to populate the database in case of disasters or other special events. Another approach that tries to provide incentives to users to report events is [11]. The authors propose a mechanism that tries to reduce the emission of carbon dioxide by reducing simultaneously the waiting time of tourists and travellers of public transportation and enabling travel sharing. However, they cannot detect, at the time of occurring, the events taking place. Other research works, e.g., [22], try to augment service provision by recognizing as well the importance for travelers to be served - in an efficient manner - by public transportation. The system developed discourages the travellers to use a specific bus stop when they have to wait for a long time. It is using a crowdsourcing approach through mobile phones, as well as various build-in sensors to sense and report the lightweight cellular signals and the surrounding environment. In our case, NaviSoc does not need to provide incentives on users to populate events, since it automatically identifies them using social streams. Moreover, it can be used both indoor and outdoor using state of the art mobile technologies.

Furthermore, Google proposed a solution to the problem of traffic jam by adding relative routing recommendation features to its Google Maps service [2]. However, despite the usefulness of the information provided by such a service, the service offered is far from satisfactory to the bus travelers. For example, the schedule of a bus may be delayed due to many unpredictable factors (e.g., traffic conditions, harsh weather situation). In addition, Google also reports that it is cost-inefficient to install specific equipment (IC Card Reader) in all vehicles.

Ivannikova [15] is interested in resource reservation in wireless networks and uses sophisticated techniques to achieve it. She emphasized on predicting the location of users in order to handle network congestion. However, she cannot be characterized as privacy-conscious as she collects personal data about application events, bluetooth devices, calendar entries, calls log, and contact entries among others. Megally proposes a system [18] that extracts tracffic information from Twitter in order to be used in route planning. However, this system does not take into consideration the user context or the existence of social events. Although Hasby et al. [13] use the account of a potentially trusted source for the data mining of their system, the additional information retrieval based on a hashtag used leads to a high probability of inaccurate and irrelevantly obtained information. Additionally, their system does not take into account the user context (e.g., interests, budget, location) in order to suggest points of interest and events that the user might be interested in. Close to our paper, SuperHub [10], [9] is a project that tries to exploit social streams, in order to enable dynamic routing mainly for environmental purposes (e.g., carbon dioxide emission reduction). However, it is still under development without any visible public result.

To the best of our knowledge, NaviSoc is the only platform currently available, which contributes to both social and business welfare. NaviSoc is a novel, event-dection, and recommendation system, having a goal to suggest dynamic routes to commuters based on their context and current circumstances by taking advantage of Twitter. Simultaneously, shop owners benefit from NaviSoc as they promote their store via our system (store's location is a parameter when NaviSoc applies dynamic routing).

VI. Conclusions and Challenges

In this paper, we argue that social streams and user context can be exploited for real-time navigation. To this direction, we designed and implemented NaviSoc. NaviSoc uses a client-server architecture. The client is installed at the user's smartphone and it communicates with the central server. The client identifies a) the user context using the applications already installed in the smartphone and b) the user location using smartphone sensors, and communicates with the server. The server identifies events relevant to the user, in her nearby location, and dynamically proposes optimal navigation paths.

Clearly, there are several challenges yet to be resolved. For example, as the tweets that we handled were written in the Greek language, characterized by many types for each word (e.g., different grammatical cases for nouns), we face a challenge to detect and pinpoint specific events and locations. For instance, many tweets include the world "Iraklion" in the possessive case. Thus, GeoPlanet Yahoo web service cannot return coordinates for the city of Heraklion and NaviSoc misses the specific event, taking place in the city of Heraklion, Crete, Greece. In this stage of the development of NaviSoc, we manually replace the possessive case of "Iraklion" as in the nominative case. However, in our immediate future plans is to implement an advanced Semantic Interpreter, identifying homonyms, synonyms etc. for the Greek language, in order to detect the different types of each word.

In addition, some Twitter users, who represent venues, post only the title of the event that they host, implying its location for reasons of simplicity. To address this issue, we could employ Google search to identify the location or use a static databases of events. Our goal is to contribute to the optimum dynamic routing of user. Furthermore, it is difficult to recognize metaphorical meaning, a common phenomenon for sport news and consequently their social streams. To tackle this phenomenon, we plan to compare the content of many social streams reporting the same piece of news in order to take advantage of different writing style of its news portal. We believe that it is difficult to detect events for an enormous region (e.g., country) without a training phase because of the existence of a place with a specific name in more than one city. Practically, as we store many useful tweets in our database, we can tackle this problem by using sophisticated machine learning algorithms.

Obviously socially enchanted real-time navigation is becoming more and more important and several challenging issues remain to be investigated in near future.

REFERENCES

[1] 1stloc – provisioning location as a first-class citizen of mobile os. http://research.microsoft.com/en-us/projects/1stloc/default.aspx. Accessed: 2014-08-12.

[2] Google maps. https://maps.google.com/. Accessed: 2014-08-12.

[3] jgeoplanet library. http://www.winterwell.com/software/jgeoplanet/. Accessed: 2014-08-12.

[4] Kort game wiki. http://wiki.openstreetmap.org/wiki/Kort_Game. Accessed: 2014-08-12.

[5] Openstreetmap. http://www.openstreetmap.org/#map=1/84/140. Accessed: 2014-08-12.

[6] Smartphones: So many apps, so much time. http://www.nielsen.com/us/en/insights/news/2014/smartphones-so-many-apps--so-much-time.html. Accessed: 2014-08-12.

[7] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011.

[8] J. Bosy, J. Kaplon, W. Rohm, J. Sierny, and T. Hadas. Near real-time estimation of water vapour in the troposphere using ground gnss and the meteorological data. *Annales Geophysicae*, 30(9):1379–1391, 2012.

[9] I. Carreras, S. Gabrielli, D. Miorandi, A. Tamilin, F. Cartolano, M. Jakob, and S. Marzorati. Urban context detection and context-aware recommendation via networks of humans as sensors. In *Fifth International Workshop on Collaborative Agents, Research,Development (CARE) 2014*.

[10] I. Carreras, S. Gabrielli, D. Miorandi, A. Tamilin, F. Cartolano, M. Jakob, and S. Marzorati. Superhub: A user-centric perspective on sustainable urban mobility. In *Proceedings of the 6th ACM Workshop on Next Generation Mobile Computing for Dynamic Personalised Travel Planning*, Sense Transport '12, pages 9–10, New York, NY, USA, 2012. ACM.

[11] N. Davies, M. Lau, C. Speed, T. Cherrett, J. Dickinson, and S. Norgate. Sixth sense transport: challenges in supporting flexible time travel. February 2012.

[12] P. Fafalios, M. Baritakis, and Y. Tzitzikas. Configuring named entity extraction through real-time exploitation of linked data. In *4th International Conference on Web Intelligence, Mining and Semantics (WIMS'14)*, Thessaloniki, Greece, June 2014. ACM.

[13] M. Hasby and M. L. Khodra. Optimal path finding based on traffic information extraction from twitter. In *International Conference on ICT for Smart Society 2013*, ICISS '13, pages 1–5, Jakarta, India, 2013. IEEE.

[14] S. Hind and A. Gekker. 'outsmarting traffic, together': Driving as social navigation. *Exchanges: the Warwick Research Journal*, 1(2), 2014.

[15] E. Ivannikova. Semantic place recognition for context aware services. Master's thesis, University of Jyvaskyla, Finland, 2012.

[16] C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 155–164, New York, NY, USA, 2012. ACM.

[17] P. Liu, J. Tang, and T. Wang. Information current in twitter: Which brings hot events to the world. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 111–112, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[18] M. Megally. Information extraction from social media for route planning. Master's thesis, University of Stuttgart, Germany, 2013.

[19] J. Rantakokko, P. Handel, F. Eklof, B. Boberg, M. Junered, D. Akos, I. Skog, H. Bohlin, F. Neregrd, F. Hoffmann, D. Andersson, M. Jansson, and P. Stenumgaard. Positioning of emergency personnel in rescue operations - possibilities and vulnerabilities with existing techniques and identification of needs for future r&d. Technical Report TRITA-EE 2007:037, Royal Institute of Technology, Stockholm, July 2007.

[20] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *WWW*, pages 851–860. ACM, 2010.

[21] M. Y. S. Uddin, M. T. A. Amin, H. Le, T. Abdelzaher, B. Szymanski, and T. Nguyen. On diversifying source selection in social sensing. In *Proceedings of the 9th International Conference on Networked Sensing Systems*, INSS '12, pages 1–8, Antwerp, Belgium, 2012. IEEE.

[22] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 379–392, New York, NY, USA, 2012. ACM.