Supporting Efficient Social Media Search in Cyber-Physical Web

Lidan Shou Sai Wu †
College of Computer Science and Technology, Zhejiang University, China
{should,wusai}@zju.edu.cn
† Corresponding Author

Abstract

The cyber-physical systems (CPS) are envisioned as a class of real-time systems integrating the computing, communication, and storage facilities with monitoring and control of the physical world. With the proliferation of social media contents in the Web, a novel type of CPS applications allow users to link interesting objects in the physical world to the corresponding social media contents in the virtual world. Such new CPS applications require novel techniques to find the semantic relationships between the two worlds, and provide efficient search tools in the integrated cyber-physical world. In this paper, we propose a general framework for supporting social media search in the cyber-physical Web, or CPSS. This framework addresses the key problems identified on different schematic levels of CPSS. We report our studies on these problems and present our solutions to four different categories of searches/recommendations in CPSS, namely the model-based search, the non-model-based search, the real-time media search, and the geo-social analytics.

1 Overview

The cyber-physical systems (CPS) are envisioned as a class of real-time systems integrating the computing, communication, and storage facilities with monitoring and control of the physical world. One interesting CPS application in the mobile Internet is to provide the social media search "on the spot" with regard to the physical world that a user sees, or literally WYRIWYS (What-You-Retrieve-Is-What-You-See). For instance, a user viewing the Statue of Liberty through her smartphone camera can search for either the latest tweets about the statue or its history from Wikipedia.

Social media search in CPS, or Cyber-Physical Social-media Search (CPSS), poses new technical challenges against the database and information retrieval community. First, various social media data, in the form of textual documents, tweets, images, videos etc., have to be captured and organized by a unified framework, which enables efficient access by CPSS applications. As an example, in [1] [6], new retrieval models are proposed to extract structured data, such as names and addresses from the social archives, to rebuild the social relationship and support efficient search. Second, some key techniques are required to support WYRIWYS, such as retrieving objects by visibility [10] [7], searching with spatial information [15] [12], and linking objects to their semantic tags etc. Third, different applications adopt different search philosophy. Thus, the ranking and recommendation algorithms should be redesigned as well [2] [8] [13].

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

Figure 1: The S4 document browser showing an object and a tweet box

Some people consider CPSS as Search with Augmented Reality (AR) interface. However, this opinion is only partly right, as CPSS is much more than Augmented Reality Search. We believe that CPSS is uniquely identified by the following properties:

- CPSS is tightly coupled with physical attributes of objects.
- CPSS requires real-time search over multi-media data.
- CPSS is highly relevant to social networks.

In this paper, we introduce a unified framework called S4 (literally for Sensor-enhanced Social media Search System) for supporting CPSS. This framework eases the development of various applications using a generic hyper-media markup language which we call Cyber-Physical Markup Language (CPML). Users can navigate in the physical world, which is integrated with documents written in CPML, using an Augmented-Reality browser. S4 has an infrastructure supporting the basic navigation and interaction with CPML documents which are bound to physical locations or objects. Figure 1 illustrates the mobile browser screen of document "Library" containing a place-of-interest (POI) and a real-time tweet box.

As a part of the framework, we have implemented in S4 some of the key modules to support WYRIWYS, such as searching by visibility and real-time tweet searching/summarization [5] [9] [11] [3]. Applications in our framework can leverage these modules to enhance their own functionalities. We will report our design philosophy and techniques used in developing these modules. We will also give our vision and suggestions on some relevant research problems.

2 Overview of the S4 Framework

Cyber-Physical Social-media Search (CPSS) differs from conventional search in many aspects. In what follows, we discuss these differences at several schematic levels.

(1) At the *Query level*, traditional search accepts keywords or images as the query input, while CPSS is required to support more complex query types with novel search semantics. For instance, one distinguishing feature of CPSS applications is the need for Augmented Reality (AR) user interface as an attempt to support WYRIWYS.

(2) At the *Data Access/Processing level*, CPSS searches virtual "documents" which combine the contents of the static HTML documents, the data attributes collected from physical objects, and the social media associated with the corresponding objects. This requirement poses technical challenges in data access and processing.

(3) *Infrastructure level* Apart from the internet, CPSS also requires certain infrastructure support, such as indoor-space localization and sensor network communication.

In the remainder of this section, we propose a general framework called S4, literally for (Sensor-enhanced Social media Search System) as our solution to CPSS. We shall focus on the *first two levels*

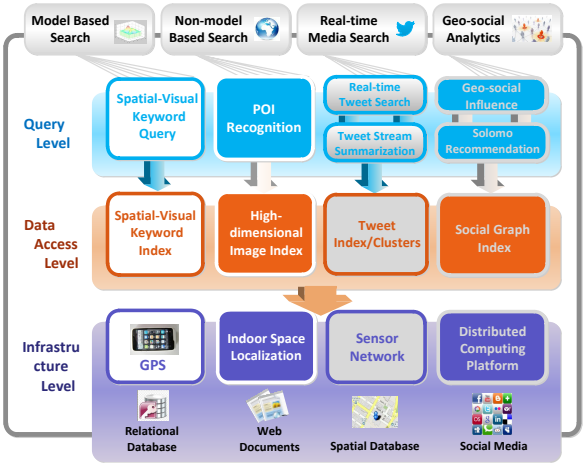

Figure 2: Schematic View of S4 Framework

and report our recent studies on these topics. The third level is necessary for the implementation of the S4 framework. However, detailed discussion of CPSS infrastructure is out of the scope of this paper. Apart from the mentioned work, we believe there are still countless open questions to be answered and problems to be solved.

Figure 2 shows a schematic view of the S4 search engine. To help understand the functionalities of proprietary components in S4, we shall introduce these components in different vertical (application) categories, depicted in four columns in the diagram.

Category 1 (Model-Based Search) One prominent requirement in CPSS is the user's ability to search for documents associated with the physical objects that are captured by the phone's camera. To support such query, namely*WYRIWYS*, we need a special index structure which can retrieve objects not only by textual relevancy but also by user's physical visibility to these objects.

Category 2 (Non-Model-Based Search) S4 provides multi-modal POI recognition functionality at the query level. This functionality relies on the support from both spatial indexing and object matching with computer vision techniques. The latter is built on top of a *High-Dimensional Image Index*.

Category 3 (Real-Time Media Search And Analysis) The social media generated from mobile devices can be linked to the physical objects, due to the availability of various device sensors. Compared to the conventional Web search, social media search poses new challenges as we need to merge the results from the physical world and the virtual world in real-time.

Category 4 (Geo-Social Analytics) Search and recommendation in CPSS should consider the users' social relationship and geo-locations. S4 includes a number of modules supporting geo-social analysis. The *Geo-Social Influence* module computes at scale the Geo-Social Influence of historical events or places recorded in large geo-social networks. Specifically, the module can tell which events (or places) are of global attraction, while the others are more suitable only for the local people.

3 Modules of S4

3.1 Model-Based Search

Model-based search assumes that the physical objects have their virtual geometric models stored in the database of the CPSS engine. S4 implements the *spatial-visual keyword* (SVK) query for processing model-based search [14]. Specifically, given a collection C of spatial keyword objects and a query q, a SVK query returns a list of k objects in C with the highest scores, where the score of each object is computed by combining its physical visibility and semantic relevancy with respect to q. The SVK query is different from simply

imposing visible judgment on distance-based spatial keyword queries, since the visual conspicuousness of objects should be quantitatively measured, based on human visual perception. For this purpose, we used a novel visibility metric, which measures the visible parts of an object in a cumulative way.

The SVK query is processed using a *Complete Occlusion-Map Retrieval* (COR) method. This method employs a hybrid index of spatial keyword objects called IR-tree [4] and works in a two-step manner.

- In the first step, a dynamic structure called *Complete Occlusion-Map* (COM) is built. This structure partitions the surrounding space of the query point into a number of angular ranges and maintains the visibility information for each range.
- In the second step, COR computes the concrete visibility for objects, as well as the tightest visibility upper bounds for IR-tree nodes. Owing to the best-first search paradigm, the search space can be effectively reduced and the top-k relevant objects are returned in an incremental manner.

The SVK query can be used to provide reality-augmented Web search for mobile users. The work presented in [14] is a preliminary study which considers the simplest form of textual relevancy and the so-called 2.5 dimensional visibility. It can be extended to more complicated metrics such as semantic relevancy and 3D space visibility.

3.2 Non-Model-Based Search

Non-model-based search refers to the case when the geometric model of objects is not available and that the sensor errors (as for location, orientation, viewing angles etc.) are fully considered for those stored in the database. One essential problem for non-model-based search is to recognize a POI being captured from a mobile phone camera.

The POI recognition module of S4 employs a *two-phase approach* to recognize the place of interest from a large, impure set of images downloaded from online photo sharing services [10]. (1) During the spatial phase, we use a *probabilistic field-of-view(pFOV)* model which captures the uncertainty in camera sensor data. Based on this model, all POIs relevant to the FOV are given a *likelihood* of being captured by the camera. (2) During the visual phase, we put forward the *SC-similarity* relying on the Sparse Coding, a technique originated from the signal processing domain. The final ranking combines an *uncertain geometric relevance* with the *visual similarity*. The most distinguishing feature of our approach is its ability to perform well in contaminated, real-world online image database. Besides, our approach is highly scalable as its implementation does not require any complex data structure.

Given the current camera parameters, the pFOV culling algorithm can quickly determine the candidate POIs which may possibly appear in the image. As a result, the cost of subsequent evaluation of geo-relevance and SC-similarity can be reduced significantly.

To compute the visual similarity, each candidate image is represented as a bag-of-visual-words column vector. Let D be the matrix where each column is the vector for a candidate image. Then the problem can be described as: Given a query image $\mathbf{x}$, can we represent it as a linear combination of other candidate images(columns in D)? This is a typical problem of sparse coding, where we try to find an optimal weight vector aiming at reproducing the query image from the candiates. The output of the solution to this problem (the weight values) are defined as our SC-similarity.

Finally, for each candidate POI, we compute a voting score as a linear combination of the geo-relevance and SC-similarity of each candidate POI. The POI with the top ranking score is taken as the final result. Experiments on densely populated areas report recognition accuracy of up to 92%.

3.3 Real-Time Media Search

The need for real-time media search comes from the explosive growth of user generated contents on the Web. In S4, we provide the real-time search and summarization modules for tweets (short texts in microblogs).

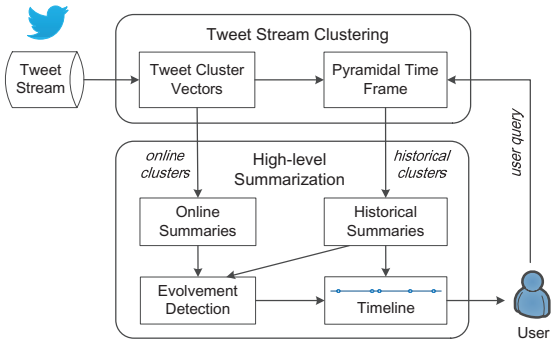

Figure 3: The Tweet Stream Summarization Module of S4

3.3.1 Real-Time Tweet Indexing and Search

In order to provide real-time search on fast arriving tweets, S4 employs the *TI*, an efficient indexing scheme which relies on inverted index [2]. For each inserted microblog, it is identified either as an important blog or a noise. Important blogs are indexed in real-time, while noisy blogs are indexed in a batch manner. To help rank the microblogs, some statistics are maintained in memory.

TI adopts the bag-of-words model, which splits each microblog into a set of keywords. The inverted index of the keywords are maintained in *TI*'s database. For a keyword, its index entry consists of a set of microblog IDs, which are sorted by their timestamps (the time when a tweet is inserted into the system). We maintain some social information in the index to support complex ranking functions.

1. In *TI*, the users of microblog are ranked based on their proprietary PageRank values in the user graph of microblog system. The PageRank value of the publisher is appended to the records of the corresponding microblog.

2. *TI* organizes the microblogs as a topic tree based on the reply/comment relationship. So we can rank a microblog based on the popularity of the whole topic. The tree ID is then maintained by the index as well.

3. Finally, we maintain the TF (Term Frequency) value for the keywords of each microblog as well. The basic TF/IDF ranking function can be applied.

4. The ranking function in *TI* combines the timestamps, the TF/IDF value, the PageRank and the topic popularity. Based on the above information, we can quickly classify a microblog as an important or noisy microblog and adopt different indexing scheme accordingly.

3.3.2 Real-Time Tweet Summarization

To support real-time summarization of fast arriving tweet streams, S4 employs a scheme called *continuous tweet summarization* [11]. The advantage of the scheme is that, given a topic-related tweet stream, one can (i) continuously monitor the stream and produce a continuous timeline which grows by time. (ii) A range timeline can also be provided to illustrate the big picture of topic evolution during some period. (iii) Users can ask for summaries of arbitrary time durations (like drill-down or roll-up summary). Such functionalities would not only facilitate easy navigation in topic-relevant tweets, but also support a range of data analysis tasks such as instant reports or historical survey.

The structure of the tweet stream timeline module is depicted in Figure 3. The module consists of two main components, namely a *Tweet Stream Clustering* component and a *High-level Summarization* component. In the *Tweet Stream Clustering* component, we design an efficient *tweet stream clustering algorithm*,

an online algorithm allowing for effective clustering of tweets with only one pass over the data. This algorithm uses two data structures to keep important tweet information in clusters. One is a compressed structure called *Tweet Cluster Vector (TCV)*. TCVs are considered as potential sub-topic delegates and maintained dynamically in memory during stream processing. The other is a *Pyramidal Time Frame (PTF)*, which is used to store and organize cluster snapshots at different moments. The PTF allows historical tweet data to be retrieved by any arbitrary time durations.

In *High-level Summarization*, we generate two kinds of summaries: online summaries and historical ones. The summarization module also contains a topic evolving detection algorithm, which consumes online/historical summaries to produce continuous/range timelines.

3.4 Geo-Social Analytics

The S4 engine also provides functionalities for geo-social analytics. We present the techniques used in two modules for geo-social analytics in S4. One is for analyzing the geo-social influence of users and events. The other is for recommending items in SoLoMo application.

3.4.1 Geo-Social Influence

This module conducts an in-depth analysis on the geographical and social correlations among SoLoMo users for different events. Assuming that each user in the geo-social network is associated with a number of events, the basic queries to be answered are:

- *User Influence* In a geo-social network, how can we measure the geo-social influence of one user to the others?
- *Influential Events Discovery* Given a set of events in the network, which ones are the influential events?

In our solution, we provide a unified user influence metric which combines social proximity and geographical mobility features of geo-social network users [15]. On the social side, we use a modified version of the hitting time measure, named *penalized hitting time* (PHT), to quantify the social proximity between LBSN users. Hitting time is a random-walk-based graph proximity measure which has been shown to be effective for link prediction, query suggestion, graph clustering, and so on. However, it is sensitive to long paths and tends to benefit popular entities. Our PHT measure intrinsically avoids this drawback. On the geographical side, we model the geographical influence with regard to distance by the *power law distribution*.

For efficient computation of PHT, our solution uses two approximate algorithms, namely the *global iteration* (GI) and the *dynamic neighborhood expansion* (DNE) algorithms. Both algorithms work efficiently when computing PHT, and meanwhile ensure tight theoretical error bounds. In particular, the DNE algorithm can compute PHT in a constant time regardless of the network size.

Relying on the user influence metric, the influence of an event can be measured by aggregating the influences of different users, with two specific aggregate functions, namely MAX and AVERAGE. We employ the sampling technique to avoid computing geo-social influence for each user when estimating event score, and adopt the *threshold algorithm* to efficiently retrieve the top-K influential events.

3.4.2 SoLoMo Recommendation

In S4, to measure the similarity between users, we use a new metric called *co-space distance* which considers both the user distances in the real world (physical distance) and the virtual world (social distance). The challenge of introducing such a hybrid distance is two-fold:

First, computing the social distance between all pairs of users is costly, even in an offline manner. In our module, we use an efficient social distance computation algorithm based on the parallel processing framework of MapReduce. The results of the MapReduce jobs are indices for the social distances between

users (social index), which are inserted into a key-value store, such as Hbase[1]. To reduce the index lookup overhead, top social distances are kept in the adaptive cache.

Second, if user u_0 issues a friend recommendation query, the query engine needs to look up two indices, the location index (e.g., the R-tree index) and the social index. Because the users update their geo-locations continuously, we have to dynamically compute the co-space distances between u_0 and other users at query time. Such query processing incurs high I/O access costs. Therefore, we adopt two techniques, a progressive query processing approach and an adaptive caching approach, to optimize the kNN recommendation algorithm.

The recommendation process involves the following steps: Given a query from the mobile user, the query engine exploits the R-tree index and the social index to retrieve the distances between users. Before the two types of distances are returned to the query engine, they are merged into the co-space distances by SVM model, which is trained from the human workers in the Amazon Mechanical Turk [2]. The engine will rank the users by their co-space distances and generate the top-K users as the query result.

4 Conclusions

In summary, we presented in this paper a general CPSS framework called S4. We focused our discussions on the design and implementation of the key modules which support four different categories of queries and recommendations in CPSS, namely *model-based search*, *non-model-based search*, *real-time media search*, and *geo-social analytics*. There certainly remain vast amount of space to be explored in the mission to build up a CPSS system. We briefly discuss the challenges and open questions to face when developing CPSS applications, and our views of possible research directions.

(1) For model-based search, an essential problem is the sensor inaccuracy. For outdoor-space objects, the geometric models acquired from sensors or other sources are almost always error-prone. As for indoor-space objects, the localization of objects could be a major problem which is still being heavily studied. We do expect disruptive technology on the infrastructure-level to address this problem.

(2) The non-model-based search produces poor performance when the visual database contains impure images which include noises irrelevant to the POI being recognized. The proposed sparse-coding technique only provides a limited solution to this issue. Therefore, we expect stronger computer vision techniques to play an important role in this regard. For example, due to the large number of images available for each POI, it is possible to detect for each image the salient object of interest. As a result, the object recognition can expectedly achieve higher accuracy.

(3) For real-time media search/recommendation, the run-time performance is still a major challenge as the "big-fast data" grows further in both scale and speed. The recent developments in big data processing on new software and hardware platforms (such as GPU clusters) may shed some lights on this issue.

Apart from the above challenges, there are also needs for novel queries and recommendation algorithms from new CPSS applications. We hope this paper would solicit attention from the research community to CPSS which we envision as a future paradigm for mobile Web search.

References

[1] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 183–194, 2008.

[2] Chun Chen, Feng Li, Beng Chin Ooi, and Sai Wu. Ti: an efficient indexing mechanism for real-time search on tweets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 649–660, 2011.

[1] http://hbase.apache.org/

[2] http://aws.amazon.com/mturk/

[3] Xing Chen, Lin Li, Guandong Xu, Zhenglu Yang, and Masaru Kitsuregawa. Recommending related microblogs: A comparison between topic and wordnet based approaches. In *AAAI*, 2012.

[4] Gao Cong, Christian S. Jensen, and Dingming Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[5] Sanda M. Harabagiu and Andrew Hickl. Relevance modeling for microblog summarization. In *ICWSM*, 2011.

[6] Chia-Jung Lee, W. Bruce Croft, and Jin Young Kim. Evaluating search in personal social media collections. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 683–692, 2012.

[7] Chen Liu, Sai Wu, Shouxu Jiang, and Anthony K. H. Tung. Cross domain search by exploiting wikipedia. In *ICDE*, pages 546–557, 2012.

[8] Rinkesh Nagmoti, Ankur Teredesai, and Martine De Cock. Ranking approaches for microblog search. In *Web Intelligence*, pages 153–157, 2010.

[9] Brendan O'Connor, Michel Krieger, and David Ahn. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*, 2010.

[10] Pai Peng, Lidan Shou, Ke Chen, Gang Chen, and Sai Wu. The knowing camera: recognizing places-of-interest in smartphone photos. In *SIGIR*, pages 969–972, 2013.

[11] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: continuous summarization of evolving tweet streams. In *SIGIR*, pages 533–542, 2013.

[12] Shuangyong Song, Qiudan Li, and Hongyun Bao. Detecting dynamic association among twitter topics. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 605–606, 2012.

[13] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 35–44, 2011.

[14] Chao Zhang, Lidan Shou, Ke Chen, and Gang Chen. See-to-retrieve: efficient processing of spatio-visual keyword queries. In *SIGIR*, pages 681–690, 2012.

[15] Chao Zhang, Lidan Shou, Ke Chen, Gang Chen, and Yijun Bei. Evaluating geo-social influence in location-based social networks. In *CIKM*, pages 1442–1451, 2012.