http://dayta.me - A Personal News + Data Recommender for Your Day

Ali Al-Mahrubi, Gianluca Correndo, Hugh Glaser, Augustinas Grieze, Stephen Lynch, Ian C. Millard, Emmanuel Munyadzwe, Tope Omitola, Igor Popov, Darren Richardson, Manuel Salvadores, Daniel A. Smith, Max Van Kleek, Yang Yang, mc schraefel, Wendy Hall, Tim Berners-Lee, and Nigel Shadbolt

EnAKTing, Electronics and Computer Science,
University of Southampton, Southampton, UK.
http://dayta.me

Abstract. http://dayta.me is a highly personal information recommender that augments a person's online calendar with useful information pertaining to their upcoming activities. To perform this recommendation, it draws upon a large collection of distributed, linked-data and Web 2.0 data sources it queries live, and provides a clear, simple user interface that regular citizens can easily use to situate useful, but formerly inaccessible, information within the context of their daily activities.

Keywords: information presentation, recommendation service, linked data

1 Introduction

The recent emergence of location-based services have started to demonstrate the usefulness of delivering information to people based upon their whereabouts and activities. But such emphasis on a person's *current* activities limits the potential utility of this information, as it is often too late for the user to make effective use of it. For example, it is of little use for a person to be informed about bad weather or traffic if they are already experiencing it first-hand, or similarly, to be informed of a colleague's inability to attend a meeting that has already begun.

Thus, it is often considerably more useful to recommend information in advance of when a user is likely to need it. A core requirement towards anticipatory information recommendation is a reliable source of predictions about the user's likely upcoming activities. One often-overlooked source of this kind of information is already widely available: online personal calendars.

For the 2010 Semantic Web Challenge, we have designed dayta.me, a personal data recommender driven by a user's personal calendar that blends (RDF) Linked Data [2] and Web 2.0 data sources to make personalized recommendations of web-based information items that the user will likely need during her day. These recommendations are based on the people she will meet, places she will visit, and the activities she will be engaged in, according to her current location and calendar. These information items are not restricted to those recommended

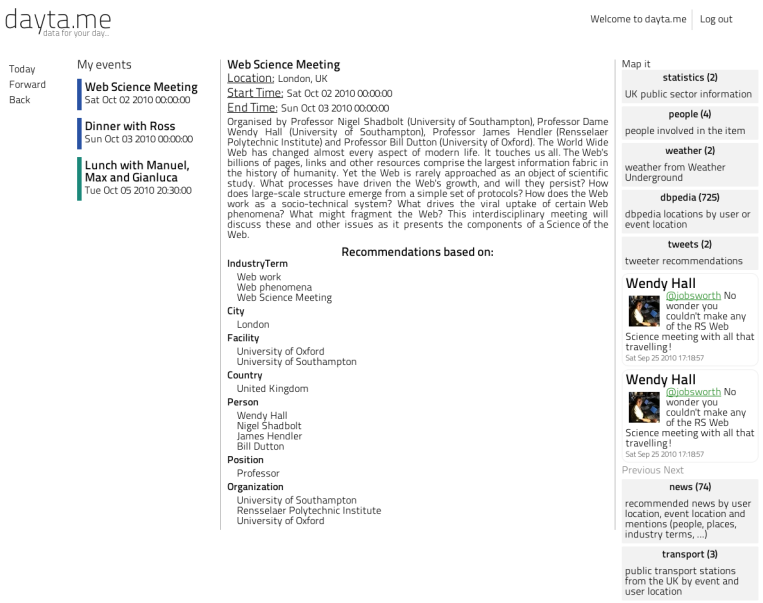

Fig. 1. dayta.me UI showing 3 columns: events, the main (focused) event, and recommended items. Recommended items are collated by type.

by traditional news and social media items such as news articles, status updates, tweets, etc. but also can be used to recommend views of arbitrary RDF information, including visualisations of statistical data, both from real-time sources (such as weather, traffic) as well as from archived, large statistical data sets, such as public sector data archived by governments (e.g., `data.gov.uk`)

The foremost aim of this project is to provide regular citizens with a view of their daily calendars that is enriched with pertinent information selected from the huge quantities of Linked Data available today. This information should aim to help them maintain better situational awareness and be more prepared for upcoming events. A secondary goal is to demonstrate how Linked Data and semi-structured (Web 2.0) data sources can be used in tandem to solve a common task: personalised information recommendation.

We first begin with a walk-through, followed by an overview of how the system works, and conclude with a brief description of the challenges encountered in achieving a distributed-knowledge-based recommendation system using a mixture of linked data and Web 2.0 data sources.

2 Scenario based walk-through

In this section, we start with a quick walk-through scenario of how a user interacts with `dayta.me`.

Mario, a new postgraduate student, is instructed by his supervisor to attend a Web Science conference in London next week. To remind himself to make travel

plans, Mario copies and pastes the event's description from the web site into his Google Calendar.

Two days before the conference, Mario opens his `dayta.me` (Figure 1) and is reminded by the entry to finalize his travel plans. When he clicks on the event in his calendar, `dayta.me` suggests a variety of information pertaining to the event itself: nearby points of interest, tweets by people hosting the conference, and news and blog articles about the conference. Interested in nearby attractions, Mario switches to the map view (Figure 2a), an alternative to the list format that spatially visualises items that have geographic coordinates. Using this view, Mario quickly notices that that the National Gallery (retrieved from DBpedia), is only one block away from the venue, but is only open during the day; so, he decides to try to visit during the first lunch break of the conference.

In addition to these items, `dayta.me` also recommends he review statistics pertaining to the conference's location, including crime rates in the area over recent years, and weather and temperature forecasts for the day in question. These are slices of government data sets (from `data.gov(.uk)`) and summaries of real-time data sources[1] that are selected and rendered into easy-to-read charts. He is dismayed to discover that certain areas near the conference have particularly high crime (Figure 2b). As the conference ends late, he decides to look for nearby hotels so he will not have to walk far.

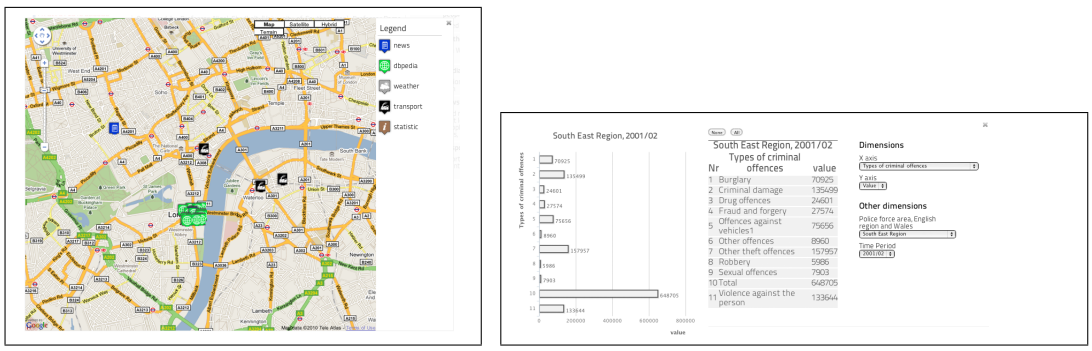

Fig. 2. (a) Map view for recommended items. (b) Example of statistical (SCOVO) interactive widget for public sector info, such as mortality rates, unemployment etc.

3 dayta.me architecture

`dayta.me` consists of two components: the dayta interface (which runs in the browser), and its server-side support: the recommender engine, a triple store, and the *fresh item fetcher*, which constantly retrieves new data items from live feeds and adds them to the triple store. These components are described next:

[1] such as from `http://data.london.gov.uk/apibeta`

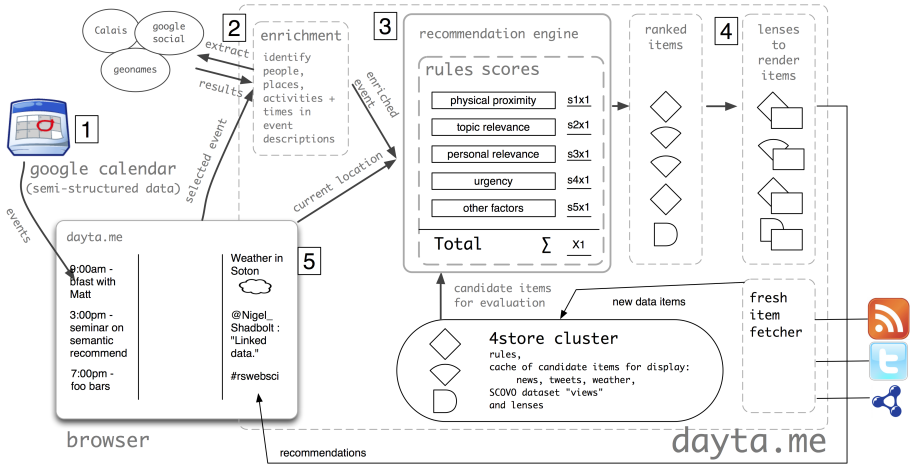

Fig. 3. dayta.me architecture and data flow, described in Section 3

1. *Retrieve personal events and user context* - When a user logs into the `dayta.me` application, browser-side code first retrieves the user's private calendars[2], and displays upcoming events in the left-hand-column. At the same time, the browser also retrieves the user's sensed location through the HTML5-standardized Geolocation API.
2. *Event selection* - When the user clicks on an event, that event and the the user's location are bundled up and transmitted to the server as *the basis for recommendation*. Once on the server, it is then transformed into RDF using the semantic enrichment process described in 5 but is never persisted; as a result, no private personal information is accumulated on the server.
3. *Recommendation* - The core recommendation process is a simple *rule-based recommender* in which modules known as rules compute scores for each candidate data item during the voting process. When such scores are computed, they are aggregated (summed) and then used to rank the items. In `dayta.me`, Those that have top scores above a threshold are passed to the lens selection engine.
4. *Lens selection* - Since the best way to display a particular information item largely depends on what it represents, `dayta.me` uses *lenses* to flexibly define how particular information items should be visually presented in the interface. Lenses, similar to those found in Fresnel, are RDF entities consisting of a pattern (which specifies the structures of entities it is designed to render), an HTML template, associated CSS styling, and (optionally) javascript controller code. Lenses are matched with data items to be presented by applying each lens' pattern (expressed in SPARQL) with each data item successively;

[2] Currently this is done using Google Calendar GDATA API, but will be expanded to other popular online calendars soon.

variable bindings in items that match are then used to populate appropriate fields in the template.

5. *Rendering* - Upon receiving the relevant data items and their corresponding lenses, the browser instantiates the HTML template fragments of the lenses with their corresponding items to produce the interactive view of these recommended elements.

3.1 Rules

Recommendation rules simply consist of a SPARQL query (that acts as the rule's *antecedent*) which selects candidate item(s) from the triple store to display, and a python expression (that serves as a *consequent/scoring function*) which yields a score (0-1.0) based on inferred relevance. At a high level, the rules we designed for `dayta.me` score items by their *location*, *topic*, *temporal* and *participant* relevance. To compute relevance along each of these dimensions, rules rely on a number of external data sources as described next. Rules may be easily added or removed simply by deleting corresponding Rule entities in `dayta.me`'s triplestore.

3.2 Data sources

`dayta.me` currently uses more than 30 data sources and services (of 17 different varieties), for three different categories, illustrated in Table 1. The first consists of sources for user context, including the user's calendar (i.e., Google Calendar), browser (for location), and, the user herself (i.e., manually supplied events). The second category consists of sources of items to recommend, including events/article feeds, tweets from Twitter, Freebase and dbpedia articles on people and entities mentioned, government PSI statistical data sets, and transportation information and points of interest from NAPTAN[3], and OpenlyLocal[4]. The final category pertains to data sources that are consulted as external knowledge-bases by rules during the recommendation process to compute relevance.

Sources of user context	**Sources of items *to recommend***	**Sources used *during recommendation***
Google Calendar (Events)	RSS feeds (e.g. BBC, NYT, xkcd..)	OpenCalais (extraction)
Browser (Geolocation)	Twitter (tweets)	Geonames (disambiguation)
Events form	dbpedia	EnAKTing Geoservice (containment/proximity resolution)
User clicks on items	Freebase (biographic profiles)	Freebase (disambiguation)
	NAPTAN (rail, bus stops)	DBpedia (topic distance)
	EnAKTing statistics datasets (mortality, crime,...)	`http://mapit.mysociety.org`
	TFL (traffic, tube status)	Ordnance Survey (postcode/disambiguation)
	Weather Service	

Table 1. Data sources used by dayta.me, categorized by how they are used.

[3] National Public Transport Access Node, `http://www.dft.gov.uk/naptan/`

[4] `openlylocal.com`

4 Related work

Among the three types of recommendation systems [1], `dayta.me` is a type of decentralized [6], knowledge-based recommender, that incorporates real-time user-contextual sources (location and calendar) similar in fashion to that described in [3].

5 Discussion and challenges

Two primary difficulties encountered in the process of building *dayta.me* surrounded the lack of suitable meta-data for use in recommendation, and the difficulty of reconciling terminological inconsistency among `dayta.me`'s sources. As described in Sec. 1, the presence of meta-data describing the *who*, *what*, *when*, *where*, etc., of information items is essential for the rules that comprise `dayta.me`'s recommendation process. While RDF provides an expressive way to represent such meta-data, few of the popular data sources offer RDF descriptions at this level. This is, in part, due to the widespread adoption of the simpler, non-RDF RSS 0.95 and ATOM news syndication formats which offer minimal structure, no flexibility, and unstructured textual values rather than URIs. But even among services that return linked-data, such as OpenCalais, the need to resolve co-referring URIs remains; few resources contain adequate (or any) `sameas` linkage to be able to easily and efficiently trace when two URIs extensionally co-refer.

Metadata enrichment To deal with the first challenge, `dayta.me` enriches provided metadata for all acquired items by running an information extraction process over all textual descriptions attached to items (and, at times, over the entire retrieved item(s), such as news article bodies). The extraction of these references is currently done using the OpenCalais named entity recognition APIs[5], which produces a summary of all the relevant lexical terms with URIs that identifies them, along with term types.

Ambiguity resolution Resolving whether two entities co-refer in the absence of explicit sameAs linkages is more difficult. We use a two-fold solution to this problem; the first using `sameas.org`, a repository for sameAs linkages that we have been building using contributions from both data set authors and users [5]. The second is a set of domain-specific approaches we crafted to make things work, such as for resolving place-name references using geographic relationships among co-occuring places. For example, a news article containing the text "Oxford, United Kingdom" parsed by OpenCalais yields two URIs: one for "Oxford"[6] and one for "United Kingdom". The difficulty is that, instead of representing

[5] `www.opencalais.com`

[6] Oxford: `http://d.opencalais.com/genericHasher-1/3966df4f-\`
`\3a0a-3de2-b179-48423c2d7c72`

the city of Oxford, in Oxfordshire, UK, dereferencing the first URI reveals that it represents *all* cities named *Oxford* worldwide – ignoring the proximity of the second mention in the orignal text. Thus, for this case, `dayta.me` checks location URIs returned by OpenCalais and, when ambiguous, constrains intepretation based on geographic proximity to co-mentioned locations.

To recommend statstical datasets, `dayta.me` first applies the `mapit.mysociety.org` APIs to reverse-geocode SCOVO location dimensions of data sets to Ordnance Survey (OS) URIs. The EnAKTing geoservice[4] is then used to infer the URIs for all containing regions, which are each compared to the coordinates of the location(s) of interest.

6 Conclusion and Ongoing work

We created `dayta.me` to determine whether information in the Web of Linked Data could be made more relevant to people by situating it within the context of their daily activities. But we see `dayta.me` also as a starting point towards determing how linked-data could be used in the recommendation process – specifically, towards building better, multifaceted proactive information systems that can more precisely target the highly personal, potentially rapidly-changing information needs of users in ways that today's pure-collaborative filtering methods alone cannot. In our ongoing work, we are measuring the effectiveness of our semantic recommendation approaches towards recommending relevant information to real users using both qualitative, subjective user-experience metrics and quantitative measurements of use.

7 Acknowledgements

This work was supported by the EnAKTing project funded by the Engineering and Physical Sciences Research Council under contract EP/G008493/1.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering pp. 734–749 (2005)
2. Berners-Lee, T.: Design issues: Linked data. `http://www.w3.org/DesignIssues/LinkedData.html` (2006)
3. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
4. Correndo, G., Salvadores, M., Yang, Y., Gibbins, N., Shadbolt, N.: Geographical service: a compass for the web of data. In: WWW2010 Workshop: Linked Data on the Web (LDOW2010) (April 2010)
5. Glaser, H., Jaffri, A., Millard, I.: Managing co-reference on the semantic web. In: WWW2009 Workshop: Linked Data on the Web (LDOW2009) (April 2009), `http://eprints.ecs.soton.ac.uk/17587/`
6. Heitmann, B., Hayes, C.: Using linked data to build open, collaborative recommender systems. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence (2010)

8 Appendix

Main criteria	**Does dayta.me meet it?**
The app has to be an end-user app, (i.e. an app that provides a practical value to general Web users)	Absolutely. dayta.me was designed for daily-use by non-specialized Web users.
Info sources used should be under diverse ownership or control, should be heterogeneous (syntactically, structurally, and semantically), and contain substantial real world data.	Yes. As described in Section 3.2 dayta.me relies on 30+ data sources many under different ownership. Some are linked-data sources while others are "Web 2.0" semistructured sources w/ APIs. All data sources are "real", queried live and only cached for performance.
The meaning of data has to play a central role. "Meaning" must be represented using Semantic Web technologies.	Yes. As described in Sec. 5, the core recommendation process relies on semantics extracted from candidates items (i.e.topics, places, persons of relevance). Furthermore the relevance between entities (topics, entities) is assessed using semantic distance.
Data must be manipulated/processed in interesting ways to derive useful info; this processing has to play a central role in achieving things that alternative technologies cannot do as well, or at all.	Yes. Semantic technologies are used to enabled widely-heteogeneous information to be leveraged towards generating highly personalized recommendations driven by extremely sparse data (e.g., a single user's calendar events and location). This would be difficult with standard statistical (collaborative-filtering) approaches. (All pattern-directed processing of the system is done using SPARQL queries).
Additional desirable features	**How dayta.me meets it**
The application provides an attractive and functional Web interface (for human users)	Yes. We followed a user-centered design process, working with expert UI evaluations to ensure dayta.me would be appropriate for regular end-users. Features such as the "why was this recommended" explanations were added to make the system more transparent and understandable.
The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all the data that is currently published on the Semantic Web.	Yes. 1) dayta.me's open architecture permits new data sources and info types can be integrated easily. E.g., the use of lenses, allows views for new types to be seamlessly integrated. 2) dayta.me's back-end supports scaling to a large quantity of data sources and users. The typical bottleneck, the triple store, is high-performance 4store cluster, and the front-end recommendation servers scale horizontally.
Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained.	We are planning a deployment (to 100s of users) in October 2010.
Novelty, in applying semantic technology to a domain or task that have not been considered before	While this clearly derives from earlier *semantic recommenders*, it is unique in several ways: 1) relies on both Web2.0 and linked data sources in multiple places, 2) it is driven by both user context and their calendar 3) generates customized, appropriate views of SCOVO data sets.
Functionality is different from or goes beyond pure info retrieval	Yes. Dayta.me extends into the space of dynamic proactive/just-in-time retrieval systems.
The application has clear commercial potential and/or large existing user base	Yes, an extremely large potential user base: people who use online calendaring systems and news aggregation services (and wish to combine the two!).
Contextual info is used for ratings or rankings	Yes, the user's location and calendar.
Multimedia documents are used in some way	Videos or music are regularly recommended in feeds.
There is a use of dynamic data (e.g. workflows), perhaps in combination with static info	Dynamic and static data are both used throughout the system. For example, dynamic sources: user's location, weather, traffic, tweet and news streams. Static data: the user's calendar, data.gov.uk, NAPTAN statstics. See Sec 3.2.
The results should be as accurate as possible (e.g. use a ranking of results according to context)	Accuracy is determined by disambiguation process and rule performance, which, while not perfect, is acceptable for the app.