Digital Me: Controlling and Making Sense of My Digital Footprint

Mats Sjöberg[1], Hung-Han Chen[23], Patrik Floréen[12], Markus Koskela[1], Kai Kuikkaniemi[2], Tuukka Lehtiniemi[2], and Jaakko Peltonen[24]

[1] Department of Computer Science and Helsinki Institute for Information Technology HIIT, University of Helsinki
mats.sjoberg@helsinki.fi
[2] Department of Computer Science and Helsinki Institute for Information Technology HIIT, Aalto University
[3] Media Lab Helsinki, Aalto University
[4] School of Information Sciences, University of Tampere

Abstract. Our lives are getting increasingly digital; much of our personal interactions are digitally mediated. A side effect of this is a growing digital footprint, as every action is logged and stored. This data can be very powerful, e.g., a person's actions can be predicted, and deeply personal information mined. Hence, the question of who controls the digital footprint is becoming a pressing technological and social issue. We believe that the solution lies in human-centric personal data, i.e., the individuals themselves should control their own data. We claim that in order for human-centric data management to work, the individual must be supported in understanding their data. This paper introduces a personal data storage system Digital Me (DiMe). We describe the design and implementation of DiMe, and how we use state-of-the-art machine learning for visualisation and interactive modelling of the personal data. We outline several applications that can be built on top of DiMe.

Keywords: personal data management · human-centric personal data · knowledge work · text analysis · distributed representations · interactive machine learning

1 Introduction

Today, much of our daily professional and private lives are mediated through digital technology. This means we are in constant interaction with information systems, sometimes even without realising it. Most – if not all – of this interaction is logged and often stored for a long time. This massive *digital footprint* can be analysed to gain insight into a particular persons behaviour, personal preferences and needs, and even predict future actions. Such knowledge could be used to design a new breed of interactive systems, in which computers would do what they are best at (data processing and statistical modelling) to support humans doing what they are best at (creativity and sense-making). For example, a

proactive computer system could be designed that can anticipate the user's needs based on previous behaviour. A memory expander system could be designed to help recall previous events, e.g., what was discussed in the meeting last week or that interesting article you read a few days ago. A tool for analysing your daily behaviour at work could help you better manage your work time.

Unfortunately, the collected *personal data* is typically not available for the individuals themselves, instead it is often collected and stored in a centralised manner by one of the big Internet companies, such as Google, Apple or Facebook. The use of this data is restricted by the functions provided by the owners of these centralised points of collection. In fact, the data may not even be used in the individual's best interests, as it is controlled by another entity with sometimes conflicting interests. Furthermore, the user may not even know what data is being collected and stored about her. Finally, the collected digital footprint is often not in a single location, but different parts are locked down into several proprietary silos, which do not share data between each other. In order to get the full benefits from the data collection, there is a strong incentive for consumers of these services to use only a single company's tools, as most of the data would then be collected in a single location. This is obviously detrimental to market competition and innovation, as the user cannot easily take her data and move to a competing provider. However, even in this single-vendor scenario, the utilisation of the personal data is ultimately controlled by the vendor, not the user.

The key to unlock the benefits of personal data for the individual, while avoiding the pitfalls of vendor lock-in and privacy nightmares, lies in human-centric *personal data storage (PDS)* systems. The term MyData [21] refers to this paradigm shift in personal data management and processing that seeks to transform the current organisation-centric system into a human-centric one. In this approach, personal data is a resource that the individual controls, and external services can use this data only to the extent that the user gives them access. Further driving this development is recent EU legislation [7] according to which individuals must have machine readable access to all data about them.

In this paper we present our implementation of a PDS system: the *Digital Me (DiMe)* platform. DiMe is a personal data storage system, which collects the individual's digital footprint from personal computing devices, and whose design is focused on enabling different kinds of machine learning and information processing applications to operate in the user-controlled private data repository. The interplay between interactive manipulation and automated analysis is crucial to enable efficient management of large amounts of personal data, and DiMe supports both interaction and automated modeling at numerous points within the system. DiMe was designed especially for knowledge work applications, however, the design is not limited to knowledge work and can be applied also in other kinds of personal data management scenarios, such as e-commerce, personal training, well-being, home automation and education. The DiMe platform is available as free and open source software and can be downloaded from http://reknow.fi/dime/.

2 Related Work

Comprehensive recording of one's personal media and communication has been a long-lasting aspiration. Already in 1945, Vannevar Bush had a vision of a mechanised device, *Memex*, that would store all read books, records, and communications, and enable quick consultation of the recorded material [4]. There have been several projects attempting to fulfill the Memex vision. In *MyLifeBits* [12], the goal is to digitise all personal and professional information, an activity nowadays commonly referred to as *lifelogging* [14]. *Stuff I've Seen* [9] focuses on re-using the recorded information using a single index for all pieces of information (emails, web pages, documents, calendar entries etc.) on the user's computer.

Two main approaches have been proposed to enable human-centric control of personal data. The first approach is to centralise the storage of the data itself. With this approach, the scattering of data is solved by providing individuals with a personal data storage service within which they accumulate data from various sources. The personal data storage system OpenPDS [8] is one such initiative. It is focused on the aggregation and storage of specifically log-type, large-scale behavioural metadata, such as locations or web searches, and it aims to provide its users with the possibility to give fine-grained access to such metadata. Rather than provisioning access to the raw data as such, OpenPDS includes a questions-and-answers feature intended to allow services to ask questions that are responded to based on metadata. Another example of this approach is the digital.me[5] EU project, which focuses in particular on collecting data from social web services. Also commercial developers provide personal data storage services. Digi.me[6] and Meeco[7] are proprietary personal data repositories, whose aim and approach is to become a marketplace for personal data, via which their users would be able to supply personal data to usages they deem beneficial. Cozy Cloud[8] is an open source personal cloud service, whose approach is to bring the services and analytics to the cloud with the aid of an application platform within the cloud service. Its model closely resembles a personal information management system (PIMS) as described in [1]. Another important PDS project is the Hub-of-All things[9], which is especially focused on Internet of Things applications.

The second approach to enabling the control of personal data is to focus not on containing personal data in a centralised storage, but rather on managing the flows of data between data sources and data-users. In this case, the scattering of data with disparate third parties is solved by federation of data sources [18]. The individual controls the uses of personal data by employing tools and infrastructure intended for managing permissions to access data. This is the rationale of the MyData model [21], which also has a reference architecture [10] that describes a MyData consent account system and its functions. Similar frameworks are also UMA [17] and XDI[10]. Another example is Databox [6], which is a personal networked device that contains the index of personal data and the access

[5] http://www.dime-project.eu/ [6] http://www.digi.me/
[7] http://meeco.me/ [8] http://www.cozy.io/ [9] http://hubofallthings.com/
[10] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xdi

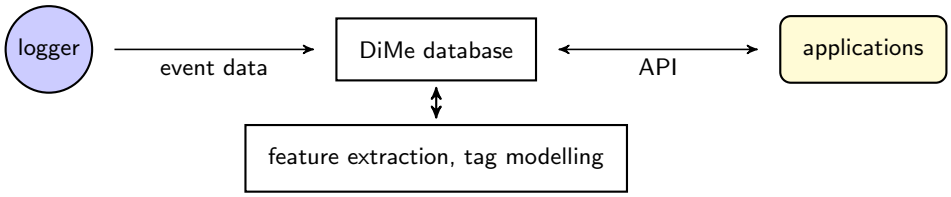

Fig. 1. DiMe architecture with loggers feeding event data into the DiMe database and applications utilising the logged data. Internally in DiMe vector representations are extracted and tag modelling is performed.

permissions. An important focus of these models is on delegation or repurposing of data to new uses. While these models are focused on permissioning rather than storing data, they may well include a PDS as one data source.

The Digital Me personal data storage system presented in this paper differs from the other PDS services in two ways. First, its development is focused on integrating with a broad set of loggers that track the digital footprint. These loggers are for example an email logger, a browser logger, a PDF reader, a desktop logger that tracks keyboard and application use, a mobile phone usage logger and a calendar logger. The second difference is that DiMe provides a representation layer to data events that is focused on providing machine learning solutions to annotating, structuring and connecting different data events.

3 Design and Implementation

The Digital Me (DiMe) system has been designed to work as an intelligent database server, which provides a programmatic interface (API) for two types of clients: *loggers* and *applications*. Internally in DiMe vector representations are extracted and tag modelling is performed (see Section 4). Figure 1 illustrates this modular architecture.

Loggers are software (or hardware + software) components that record events related to a person's actions or environment and send them to be stored in the person's own DiMe server. The primary item stored in DiMe is called *event*. Events can be, e.g., reading a document on a computer screen, the mobile phone location or the heart rate measured by a smart watch. Loggers are typically installed by the user, but then run unobtrusively in the background. The user can get an overview of what things are being recorded by checking the *dashboard*, which is the controlling web interface to the system.

DiMe applications are software components, which utilise the events stored in DiMe by the loggers. Applications typically present the user with a graphical user interface, where some part of the DiMe data is visualised and can be manipulated, or the view of the data can be modified. Examples include a time-line viewer of recoded DiMe events over time, a search system that takes into account your previously recorded interests, and a document reader that can

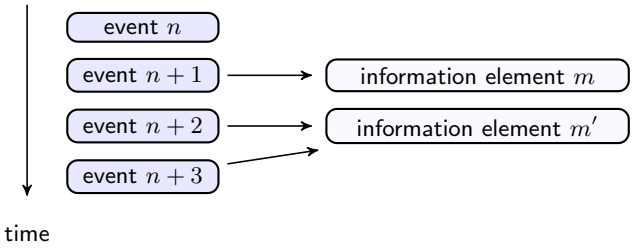

Fig. 2. In DiMe, events occur at different times and may refer to information elements.

highlight sections of the document that you have read before. We distinguish between local applications and connected applications. Local applications run on the user's own machine, while connected applications run on a server and can connect information from many different DiMe instances. In this paper, we focus mainly on local applications. Section 5 lists some potential applications, but there are numerous other possibilities as well. In practice, most applications also act as loggers, for example recording the user's interactions with the user interface.

Through the dashboard, the user can always cease logging to DiMe and also delete already recorded events. In applications relying on sharing information from the user's own DiMe with others, the user can choose what data to share. The dashboard includes a search and a filtering functionality for the data, as well as showing statistics about the data stored in DiMe.

3.1 Data Model

Figure 2 illustrates the basic data model in DiMe. The primary data is composed of events, which occur at a given point in time. The recorded event time should try to approximate the time of the actual real-world event being described, not for example when the event was recorded in the DiMe server. Some events may refer to a file or other time-independent piece of data, such as a PDF document; these we denote *information elements*. Often many events may refer to the same information element: for example the opening and closing of the same document in the computer's user interface constitutes two events referring to the same element. In addition, we define two auxiliary data types, *person* and *tag*. Person refers to an actual person, e.g., the recipient or sender of an email or a participant in a meeting. Tags are keywords or key-phrases which allow the user to describe the events or information elements in a way that is personally relevant.

The DiMe data model is based on the OSCAF Ontologies[11], which specify various aspects of personal information access and usage. The most relevant ontologies for our data model are the NEPOMUK User Action Ontology, which defines a vocabulary for describing user events on a device, and the NEPOMUK

[11] http://www.semanticdesktop.org/ontologies/

Table 1. The currently supported DiMe events

DesktopEvent	A desktop event, such as opening a document or web page.
ReadingEvent	A detailed reading event (e.g., page, paragraph or sentence).
MessageEvent	An event representing sending or receiving a message.
CalendarEvent	An event generated from the user's calendar.
BookmarkEvent	An event for adding or removing a bookmark by the user.
FeedbackEvent	An event representing user feedback, e.g., ranking a document.

Table 2. The currently supported DiMe information elements

Document	A document, e.g.,PDF, web page or word-processing file.
ScientificDocument	A scientific publication with a bibliographic record.
Message	An electronic message, such as an email or instant message.

File Ontology, which provides a vocabulary to express information extracted from various sources (e.g., files, pieces of software, and remote hosts). We have extended the ontologies for our specific purposes as listed in Tables 1 and 2. The two lists are not exhaustive, and new data types can be added as needed.

3.2 Implementation

The current implementation of the DiMe server is written in Java using the Spring framework[12]. The essential components of the core DiMe software are the API interface, database, search engine, and feature extraction framework.

The API is implemented over HTTP, largely following RESTful principles and using JSON as the data format [3]. The API currently supports uploading new objects (events or information elements), accessing, modifying or deleting existing objects, and filtering, e.g., retrieving all events from a given logger during the last three days. The API also supports text search of indexed objects, adding and removing tags, and fetching calculated features.

For the database we support various SQL implementations via Java Hibernate[13], and also MongoDB[14]. As the first stage of DiMe development has focused on running it locally, for example on the person's own laptop, the most commonly used database is the H2 embedded database[15], which makes for easy installation. For the search engine backend we currently use Lucene[16] to index both events and information elements having textual content.

The purpose of the feature extraction framework is to support the extraction of higher level features of the data objects, such as important key-phrases from text documents, visual descriptors from images, and interpretation of a low-level physiological signal (e.g., "person is stressed"). Some of these processes run in the core DiMe server and some outside as external applications, depending on the measure of access needed to the DiMe database. The extracted features are typically used as input for modelling the data, as explained in the next section.

[12] https://spring.io/ [13] http://hibernate.org/ [14] https://www.mongodb.com/
[15] http://h2database.com/ [16] http://lucene.apache.org/

4 Modelling of Personal Data

One of the central principles of the Digital Me design has been to include state-of-the-art machine learning-based modelling capabilities from the start. Some of these algorithms have to run in the core DiMe server, in particular if they need to model the entirety of the personal data or have to reprocess the entire database often (e.g., after model parameters have been updated). Other algorithms may run externally and access DiMe via the API, for example if they only process a small number of objects at a time or simply enhance or rerank the objects returned by DiMe.

There are numerous ways to model personal data that could be implemented in the DiMe platform, but in our research we have focused mainly on two aspects: *automatically generated vector representations*, and *interactive modelling of tags*. The first approach utilises state-of-the-art machine learning techniques to generate highly expressive vector representations in an unsupervised automated manner. The second approach instead attempts to include the human in the loop by using tags for interaction and modelling. After all it is *personal* data, and thus the individual user of DiMe is the one the best positioned to understand the data, and what organisation of the data makes most sense in the personal context. These two approaches are complementary – for example the vector representations can be used as part of the tag modelling mechanism – and will be explained in more detail in the following two sections.

4.1 Vector Representations

The data stored in DiMe is multimodal; it can be of many different types, such as text, images, videos and real-valued vectors representing physical measurements. However, in practice, the vast majority of the data is in textual form, or can be converted into such (e.g., speech-to-text, visual concept detection), and thus analysis and indexing textual data has been our primary focus.

The traditional approach to vector representation of text documents is to represent the content (in DiMe a single event or information element) as a weighted bag-of-tokens vector, where the vector has a fixed length and whose elements are counts (weights) of individual tokens, such as words (unigrams) or word combinations (bigrams, trigrams, keywords or key-phrases). Standard transformations, such as term frequency-inverse document frequency (TF-IDF) weighting, can be applied to emphasise rare tokens. Once such a representation exists for documents, the same representation becomes available for words: a word is represented by its weight vector across documents, and words are similar if they appear often in the same documents. More advanced transformations involve learning a statistical model that represents the prominent trends in the content, such as principal components across documents or topics of a probabilistic (hierarchical) topic model.

Another approach, which has gained a lot of interest due to recent advances in deep learning, is learning vector representations using neural networks [2, 19]. In this approach, each word is represented by a vector and several words from

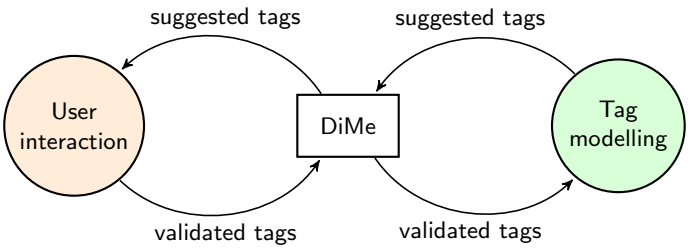

Fig. 3. The interactive tag modelling cycle, where the user validates tags suggested by the tag modelling system. The user can also add new tags manually.

the same context are concatenated or averaged to form the input to a neural network. The neural network then tries to predict other words in the context. After training this results in a vector space mapping, where semantically similar words are mapped to nearby points. For visualising and analysing personal data, we are in particular interested in representing whole documents, i.e., the stored personal information elements and events. For this we use the *Paragraph Vectors* [16] algorithm. Here, in addition to the context word vectors, a paragraph vector is included. The vector tries to capture the topic of the paragraph (or the piece of text to be represented, in our case the entire personal data item). The resulting representation has two advantages over bag-of-words methods: the ordering of the words is retained (without the data sparsity and high dimensionality problems of the n-gram methods), and the semantics of the words is taken into account. A drawback of this approach is that the individual components of the vector do not have any semantic interpretation.

Having highly expressive vector representations that capture the semantic similarity of the content is highly valuable for accessing and analysing personal data, for example for displaying similar events to the one currently being viewed in the user interface, for automatically propagating a tag to similar documents, or for visualising the global structure of the data.

4.2 Interactive Tag Modelling

Our second approach to modelling of personal data is based on using tags. Tags are keywords or key-phrases, which allow the user to describe collected events or documents in a way that is personally relevant. Each user has their own terminology and view on what is relevant. For example, a person could have a tag for a project she is currently working on, and another one for the organisational hierarchy level she belongs to. Another person may instead tag items based on the activities involved, such as programming, writing and meetings. In accordance with the conventional tagging design pattern, the tags label the events as being mapped into possibly several different potentially overlapping categories instead of the traditional hierarchical category system commonly used, e.g., in file systems.

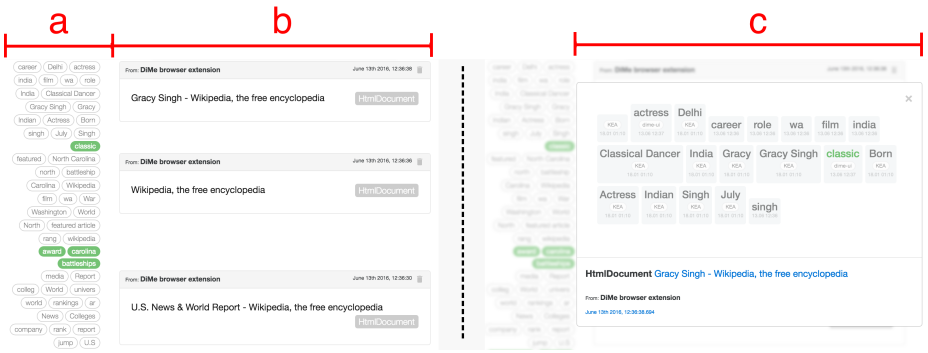

Fig. 4. The interactive tagging interface of DiMe. See text for a detailed explanation.

Figure 3 illustrates the interactive tag modelling loop, which involves both a user interaction and a machine learning component. On the left-hand side of the figure, the user interacts with the collected personal data in DiMe, either via the DiMe dashboard or via an external application. With respect to tags, this interaction involves either adding new *user generated* tags or confirming *suggested* tags. Collectively we call these *validated tags*. The validated tags are collected in DiMe together with the events they belong to, and transmitted to the tag modelling system, which is shown on the right-hand side of Figure 3.

Tags can be modelled by various machine learning algorithms, which can learn from the validated tags and the DiMe events they are assigned to in order to expand those tags to other events in DiMe. In the current DiMe system we are using the KEA [11] algorithm, which uses a naive Bayes classifier to learn an extraction scheme from the validated tags and corresponding documents. In addition to traditional classifiers, also algorithms like TagProp [13] can be used to propagate tags to neighbouring DiMe items in a suitable vector space (such as the ones discussed in Section 4.1). Correspondingly, these automatically generated tags have values attached to them related to their likelihood of belonging to that event. The most likely tags are transmitted back to DiMe together with the corresponding events; these are presented to the user as *suggested tags*, which the user, as mentioned, can confirm (or reject). The aim of suggesting tags is to expand the user generated tags, so that the user does not need to annotate all events manually. As more user feedback is gathered, the tag models are successively improved and can provide more accurate suggestions.

The interactive tag modelling approach is closely aligned with previous research on interactive machine learning [15, 22]. However, here we focus particularly on personal data, which has its unique challenges. In addition, the proposed system is more generic and could be applied using several different modelling algorithms.

Figure 4 shows two views of the DiMe user interface for displaying the collected events, and highlights several visual elements allowing interaction with tags. First, on the left-hand side is shown the basic event time-line view. The

central feature of the interface is a vertical time-line, which consists of card elements representing the individual events collected in DiMe (Figure 4-b). The associated tags are placed on the left of the event's card element (Figure 4-a). On the one hand, the tags help users identify events and the information elements they link to. On the other hand, the side-by-side display of the content and the tags enables the user to easily confirm or reject the suggested tags.

Validating a suggested tag is done by mouse-clicking and there is a visual element indicating the tag has been validated (currently the colour green has been used to represent this status, see Figure 4-a). Rejecting a tag is done by clicking on a red cross that appears when hovering the mouse cursor over the tag label (not shown here). If the user is interested in the details of a particular event, a modal window with detailed information can be opened by clicking on the event. This display is shown on the right-hand side of Figure 4: the modal information window is displayed as Figure 4-c. In our design of the tag display we have emphasised visual aesthetics since previous research has found a positive link between aesthetic visualisation and data retrieval tasks [5, 20]. We utilised the styles chosen by the well-accepted Twitter Bootstrap framework[17].

5 Applications

DiMe augments the human with a digital memory of actions undertaken. This memory can provide insight into the person's own behaviour and can be used in different applications building on this personal data. Below are some examples of applications, but generally speaking, all applications relying at least partially on referring to past events are potential DiMe applications.

Time-line Search The DiMe information can be displayed on a graphical time-line, helping to recall events and to search for particular pieces of information.

Associative Recall We may remember some partial information, but not the exact thing we look for. The information gathered in DiMe can provide cues to enable associative recall.

Proactive Search Instead of explicitly querying for information, information could be automatically provided to the user on the basis of what the user is currently doing. Previously viewed documents in DiMe can be used as a source of search results. Furthermore, the proactive system can learn about the user's interests and search preferences from the user's DiMe history.

Intelligent Meeting Room Meetings can be consistently recorded on video and audio. The meeting participants gather such information into their personal DiMes. By the participants giving explicitly access to this information, we can for instance recall what a particular person said related to a given topic in a previous meeting.

Quantified Self The person can follow his own work. As an example, information about working time and the tasks undertaken can be gathered. This can help in allocating the working time more efficiently. Consultants can automatically get information about the time to be charged to different projects.

[17] http://getbootstrap.com/

Profiling and Competence Search From the work tasks undertaken, a profile can be automatically generated highlighting the particular competences of the person. If the employees so allow, the employer can then use these profiles to optimise the composition of a group of workers, e.g., for a particular project where different complementary competences are needed.

The vector representations of Section 4.1 and tags of Section 4.2 are resources for the applications. For example in time-line search, associative recall or proactive search, vector representations can be used to find the most relevant previous documents; in competence search vector representations of employee profiles can be used to find complementary competences. Tags can be used to find which parts of the representation is most relevant.

6 Conclusions

The main contribution of this paper is the introduction of the personal data storage system Digital Me (DiMe). This serves as a platform for further research into human-centric personal data, in which individuals are in control of their own digital footprints.

However, simply gathering the data is not enough. Individuals must be supported in understanding their own data with systems that can organise and visualise the data. We propose two ways forward which utilise state-of-the-art machine learning algorithms, while supporting user interaction to modify the learned models and views of the data. Automatically extracted vector representations can be used to learn semantic relationships and visualise the structure of the collected information, or help finding related events or documents. Interactive tag modelling can learn the individual's personal categorisations from a small sample and expand them to organise the whole digital footprint.

The work is ongoing, and this paper both explains our vision and shows some potential ways forward. This paper also represents a call-for-action for interested researchers and organisations to join us in the personal data revolution!

Acknowledgments This work has been supported by the Finnish Funding Agency for Innovation (project Re:Know) and the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170).

References

1. S. Abiteboul, B. André, and D. Kaplan. Managing your digital life. *Communications of the ACM*, 58(5):32–35, 2015.
2. Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.
3. T. Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 7159 (Proposed Standard), Mar. 2014.
4. V. Bush. As we may think. *Atlantic Monthly*, July 1945.

5. N. Cawthon and A. V. Moere. The effect of aesthetic on the usability of data visualization. In *Proc. 11th International Conference on Information Visualization*, pages 637–648, 2007.
6. A. Chaudhry, J. Crowcroft, H. Howard, A. Madhavapeddy, R. Mortier, H. Haddadi, and D. McAuley. Personal Data: Thinking Inside the Box. In *Proc. 5th Decennial Aarhus Conference on Critical Alternatives*, pages 29–32, 2015.
7. Council of the European Union. General Data Protection Regulation, 2016. http://data.consilium.europa.eu/doc/document/ST-5419-2016-INIT/en/pdf.
8. Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland. openPDS: protecting the privacy of metadata through SafeAnswers. *PloS one*, 9(7), 2014.
9. S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've Seen: A system for personal information retrieval and re-use. In *Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 72–79, 2003.
10. A.-S. et al. MyData Architecture - Consent Based Approach for Personal Data Management. Published at https://github.com/HIIT/mydata-stack, 2016.
11. E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proc. 16th International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.
12. J. Gemmell, G. Bell, and R. Lueder. MyLifeBits: A personal database for everything. *Communication of the ACM*, 49(1):88–95, 2006.
13. M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proc. 12th International Conference on Computer Vision*, pages 309–316, 2009.
14. C. Gurrin, A. F. Smeaton, and A. R. Doherty. Lifelogging: Personal big data. *Foundations and Trends in Information Retrieval*, 8(1):1–125, 2014.
15. T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 3075–3084, 2014.
16. Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proc. 31st International Conference on Machine Learning*, pages 1188–1196, 2014.
17. M. P. Machulak, E. L. Maler, D. Catalano, and A. van Moorsel. User-managed access to web resources. In *Proc. 6th ACM Workshop on Digital Identity Management*, pages 35–44, 2010.
18. D. McAuley, R. Mortier, and J. Goulding. The Dataware manifesto. In *Proc. 3rd International Conf. on Communication Systems and Networks*, pages 1–6, 2011.
19. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proc. International Conference on Learning Representations Workshop*, 2013.
20. A. V. Moere and H. Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, 2011.
21. A. Poikola, K. Kuikkaniemi, and H. Honko. *MyData – A Nordic Model for human-centered personal data management and processing.* Finnish Ministry of Transport and Communications, 2015.
22. J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292, 2009.