Supporting Scholarly Awareness and Researchers' Social Interactions using PUSHPIN

Wolfgang Reinhardt, Pranav Kadam, Tobias Varlemann, Junaid Surve,
Muneeb I. Ahmad, and Johannes Magenheim

University of Paderborn
Department of Computer Science
Computer Science Education Group
Fuerstenallee 11, 33102 Paderborn, Germany
wolle@upb.de,pdkadam@mail.upb.de,tobiashv@upb.de,
jsurve@mail.upb.de,muneeb06@gmail.com,jsm@upb.de

Abstract. With the advent of Research 2.0, the way research is conducted has significantly changed. New tools and methodologies have emerged and an increasing amount of research is conducted in networked communities including the use of social networking tools. Apart from the well-known social networks, smaller and tailored social networks for researchers have emerged that are geared towards the specific needs of researchers. As more and more potentially relevant information is being made available, many researchers feel the need for awareness support in order to cope with the available amount of data. In this article we introduce the PUSHPIN application that aims at supporting researchers' awareness of publications, peers and research trends. The application is based on an eResearch infrastructure that analyzes large corpora of scientific publications and combines the extracted data with the social interactions in an active social network.

Keywords: research 2.0, eResearch infrastructure, scholarly communication, social networking, hadoop, storm, big data analysis, near-copy detection, object-centered sociality, bibliometrics

1 Introduction

In the early days, the Internet was mostly a top-down information distribution system in which only few people provided information. Users of the Internet merely consumed the information without being enabled to interact with or create own information easily. With the rise of Web 2.0, Internet usage has been revolutionized. It has enabled mankind to more easily participate in the spread of information and the participation in global discourse [9,20,21]. The different developments in Web 2.0 have resulted in a wide range of new tools and methodologies, which reshaped social interaction, distribution of news and other content as well as it fostered user participation. Applications like Facebook and Twitter not only have had impact on the worldwide social system but also influenced researchers to make applications that modernized how research is done.

The usage of Web 2.0 tools, practices and methodologies in the context of scholarly communication has been recently labeled as Science 2.0 or Research 2.0 [27,29]. Similarly, the term eResearch is used when the talk is about technologies and infrastructures to support Research 2.0, big data analysis and data sharing on a large scale. Scholarly communication is generally referred to as the publication and peer review of scientific publications. In line with [22,23] we consider scholarly communication in a broader scope and consider each social interactions and communicative activities, which is part of research cycle.

Thus, we especially consider the joint developing of ideas and the exchange of short texts, like in tweets or status updates as potentially relevant research information. Moreover, the use of social networks is considered as very relevant part of the modern research methodology. Despite the fact that Facebook has evolved to be the de-facto standard in social networking sites (SNS), there are several SNS that are tailored to the use by researchers and that help them in connecting to like-minded researcher, publications and other content.

Applications like Mendeley[1], ResearchGate[2], Academia.edu[3] or iamResearcher[4] compete with the top dog Facebook by providing features that cannot be found in the general-purpose social network. Mendeley for example focuses on the sharing and annotation of scientific documents in private or public groups. Moreover, it supports researchers in generating bibliographies and recommending publications that the research might be interested in.

However, the new way of conducting research, communicating research ideas and findings and sharing data also results in a very scattered network of potentially relevant information. Researchers are in urgent need of awareness support tools and techniques that provide detailed recommendations and hints for possible collaborators. Many of the existing approaches seem to be based on first-level metadata and collaborative filtering approaches only and this is where PUSHPIN (*Supporting Scholarly Awareness in Publications and Social Networks*) will enhance the state-of-the-art. Through the application of in-depth publication and citation analysis combined with the immense power of the social graph, PUSHPIN aims to provide better awareness support for researchers than the existing tools.

In the following sections, we present our new application called PUSHPIN and its approach for awareness support for researchers (Section 2). In Section 3, we present the implementation details for PUSHPIN and present the underlying eResearch infrastructure. We also discuss the three user interfaces for web, mobile and tabletops that PUSHPIN provides for its users. Finally in Section 4, we give an outlook on future research opportunities and present our evaluation and public release plans.

[1] http://www.mendeley.com/

[2] http://www.researchgate.net/

[3] http://academia.edu/

[4] http://www.iamresearcher.com/

2 The PUSHPIN approach for awareness support for researchers

PUSHPIN is an ongoing research project at the University of Paderborn (Germany) that aims to provide awareness support for researchers through the integration of social networking and big data analysis features. While many features of the whole approach have already been implemented and can be used, other features are not yet realized and are currently under development.

In this section we give an introduction to how PUSHPIN will help researchers to become and stay aware of their connections to other researchers and publications. In particular we describe how the social layer and the available social networking features contribute to the overall awareness of researchers (Section 2.1) and discuss the power of email notifications to keep the users engaged to visit the platform (Section 2.6). In Section 2.3, we describe how the automatic analysis of big data sets of publications is supporting object-centered sociality in PUSHPIN and how it gives insight to the relations of people and objects in PUSHPIN. Moreover, we present visualizations (Section 2.5) and recommendations (Section 2.4) that support researchers' awareness and discuss how we use mobile devices and interactive displays to access data in our ecosystem (Section 2.7).

2.1 The Social Layer of PUSHPIN

To raise awareness of an idea and to create a circle of supporters of the same, it is essential for any research idea to reach a wide audience. Social networking makes it possible to connect to potential collaborators thereby supporting the start of an incipient Research Network. Where social networking tools are often based on the *people* element, on the other hand, social awareness tools tell us a story using various *data* associated with people and helps us build a network based on such data. Often, we also find social networks that assemble around specific *objects*, which become the hub for social interactions [6]. In PUSHPIN, the objects that realize this object-centered sociality [12] are scientific publications. While PUSHPIN can identify that there is a direct connection between two researchers as they follow each other, we also provide social awareness by stating that there are x publications that both of them have cited in their own writings. This way, the system may make the researchers aware of their shared interest and common knowledge in a certain research area and may trigger a user action.

The social layer of PUSHPIN aims to support users in creating an active social network that is created by the users themselves through social interactions and conscious activities. The other parts of PUSHPIN rather contribute to a passive social network that is automatically generated by the system and that is built based on abstract information and activities such as collaboratively writing publications, working at the same institution or citing similar works [14]. Both, social networking features and social awareness support together can provide a powerful framework to support research [14,23]. The following points describe how PUSHPIN support object-centered sociality and active network constructs.

Sign-up and sign-in using existing accounts To ease the sign-up and sign-in process for users and to support them to reuse their existing social profiles as login, we enable login via Facebook, Twitter and Mendeley. Moreover, PUSHPIN gets access to the respective social graphs and can recommend friends from the other social networks that already use PUSHPIN.

User profile updates A user's profile plays an important role in getting to know the user. To name a few, it consists of information about the user's affiliations, research interests and research disciplines, which highlights the user's research areas. This information has impact on the engagement in social networks as they reflect the personality of a user. Any changes to the user's profile are presented to the followers of that user in their activity stream.

Following a user Users can follow other users to get an account of all their activities. The number of followers (users following the current user) and followees (users followed by the current user) are shown on the dashboard of a user as well as any user's profile. The number of followers of a user can be taken as quantification of the popularity and networking efforts of that user on PUSHPIN.

Status updates, likes and comments Sharing status updates is a common construct in social networking applications which allow users to share their current thoughts or their work progress. In PUSHPIN, the status updates could be used not only for sharing ideas or current readings but also for requesting help or simply sharing some news. Also, all followers of the user can like and comment on a status message, which may eventually result in a discussion of the content shared. Moreover, if a user has connected other social media accounts to his PUSHPIN account, she can automatically share the status update with all of her other accounts.

Private messaging To support non-public information exchange, all users on PUSHPIN can exchange private messages with each other. Messages are stored in conversations that multiple users can be part of. Any member of a conversation can add additional users to the conversation and each user can leave a conversation at any time.

User's activities When PUSHPIN users successfully sign in, they are redirected to their personal dashboard. A significant part of the dashboard consists of an activity stream, which is a sorted summary of activities. These activities consist of stories such as status updates of users, likes and comments on statuses, changes in profile information, users following and tagging other users, users uploading, bookmarking, rating and tagging publications, etc. In short, it tells stories of the users' interaction with other users and publications. Users can only see updates of other users, whom they follow. Apart from the dashboard, users can also see activities of a particular user on their user profile. This kind of feature is common with most of the social networking platforms including Twitter and Facebook and hence, most of the users are already familiar with it.

Uploading publications Since scientific publications are the central hub for object-centered sociality in PUSHPIN, users can upload publications to the service[5]. This may be done by selecting publication from the local computer and uploading them, or by connecting their Mendeley account to PUSHPIN. In the latter case, all the PDFs in the user's Mendeley collections are automatically imported in the PUSHPIN infrastructure. All the publications that have been uploaded to the system, are then automatically analyzed and information is extracted from them (see Section 2.3 for a detailed description of this process).

Interacting with publications All the users have access to the dedicated profiles of all the publications in PUSHPIN. On the profile, users can rate the publication and share it on other social networking sites. Moreover, users can recommend the publication to other PUSHPIN users or send the recommendation via email. Finally, users can bookmark the publication and put it in one of their collections on PUSHPIN.

Tagging objects Social tagging is one of the most prominent features of Web 2.0 [15] and is available for all kinds of objects in PUSHPIN. Users can tag publications and institutions and to classify other users they can also tag users (this is commonly referred to as people tagging [3,7,19]). When someone explores a keyword, all the users tagged with that keyword form a part of search results in researchers' list.

2.2 Publications analysis

All scientific publications that are uploaded to the PUSHPIN infrastructure[6] are automatically analyzed according to several aspects. This automatic analysis represents a series of processing steps that are executed after a publication is uploaded the PUSHPIN system.

The first and foremost step taken is to check if the publication is already in the publication corpus and/or if a full analysis has to be started. If this is not the case, the uploaded publication is inserted into HBase[7]. After that, Storm[8] is triggered for further analysis of the publication. This analysis by Storm involves activating the metadata extraction and reference extraction modules to obtain the metadata and references from the publication. The metadata, being referred to, can be the title, the author(s) and their email addresses, the authors' institutions, abstract, and keywords. For each of the references that have been cited in the publication, the reference extraction module looks for title, author(s), year of publication and publication outlet. The two modules

[5] Due to potential copyright infringements, we will only process the uploaded data in order to extract metadata from the publications. We will not, however, allow the public download of the PDFs shared with the PUSHPIN system.

[6] Currently we only process articles in PDF format. In particular, we do not process books or theses.

[7] `http://hbase.apache.org`

[8] `http://storm-project.net`

use GROBID[9] and ParsCit[10] as key software tools. If additional metadata in BibTeX or PLoS XML format is available, the modules make use of this information as well. The extracted data is then compared and combined to get the most exact metadata (similar to our approach in [26]). Alongside metadata extraction, Storm also triggers a module that creates thumbnails of each page of the uploaded publication.

2.3 Near-copy detection and publication similarities

A problem of modern science is the rising amount of plagiarism. In the digital age it has become much easier to access scientific publications and to copy content. In order to detect conscious or unconscious plagiarism we introduced algorithms to PUSHPIN, which are capable of doing near-copy detection (NCD). NCD means that correctly cited paragraphs will also be detected. To distinguish between full-text quotes and plagiarism, additional algorithms have to be used to detect plagiarism indicators. This could be done in future projects. The NCD algorithm used in PUSHPIN are inspired by the fuzzy string similarity detection algorithm described in [1].

Each uploaded paper first goes through initial text preprocessing steps before it can be analyzed by our NCD algorithm. These initial steps are used to remove irrelevant and uninteresting parts of the text and to make the different text better comparable:

Text extraction The papers are uploaded as PDF files. From these files, the text, along with the information about its position in the PDF file are extracted. This gives the exact location of a copied text in the documents it appears in.

Text cleaning The extracted text contains – for the NCD algorithm – uninteresting information, like headers and footers of the document. These lines are removed and hyphenated words are joined again.

Language detection Some algorithms need to know the language of the text as they work with trained models that are specific for one language.

Part-of-speech tagging The "Part-of-Speech" (POS) tagging determines the grammatical meaning of a word in a sentence. This information is necessary for detecting synonym groups of words later on. Moreover, POS tagging is also useful in combination with lemmatization for calculating word clouds.

Lemmatization and stemming For comparing words in our NCD algorithm, it is necessary to bring all words to the principal form, which is the same for all tenses and plural and singular forms. Lemmatization transforms words in the principal form using a dictionary algorithm. This algorithm is expensive in time and memory but the results are real words, which also can be displayed in word clouds. Stemming is an algorithmic transformation of the input word that will transform it to the stem. The stem, however, does not

[9] `http://grobid.no-ip.org`

[10] `http://aye.comp.nus.edu.sg/parsCit`

need to be a real word and thus should not be used in word clouds or the like but its calculation is very fast.

Number and stop word removal In this step, we remove unimportant elements from the text in order to reduce the complexity of the NCD algorithm computation.

Synonym detection Often, copiers try to conceal the copies by replacing words with synonyms of the word. This makes it harder to detect certain parts of a text as copied. This makes it necessary to detect synonym groups that a given word belongs to and to check all synonyms of the word for potential copies. In this step we make use of the WordNet project [16,8] and a modified Lesk algorithm [2] for distinguishing the different meanings of a word.

After the text preprocessing is finished, the NCD algorithm can calculate the similarities between all sentences of the publication and the preprocessed background corpus. This procedure is inspired by [1] but additionally incorporates the similarity between two synonym groups. Whereas the original algorithm uses a similarity of 1 if two words are equal, a similarity of 0.5 if they are in the same WordNet synonym groups and 0 in all other cases, we calculate the Wu and Palmer WordNet similarity [33] between two words if they are not equal. Additionally to the sentence-level calculation of similarities, we also compute several text-based similarity measures on a fulltext-level of all publications in the PUSHPIN corpus with respect to each other.

This computation needs very large computational power and produces a lot of similarity data. We rely on the Apache Hadoop framework to scale the computation to a cluster of computers (see Section 3 for a detailed inspection of the PUSHPIN eResearch infrastructure).

2.4 Recommendations

In PUSHPIN we use an ample number of recommender algorithms due to the following reasons:

1. The system has to take into consideration the networks that result from the extracted co-authorship information as well as the co-citation and bibliographic coupling data of publications.
2. For item-based recommendations, the system also has to employ the use of textual similarities, clustering results, author-assigned and extracted keywords as well as user tags.
3. Also, the system is capable of tracking user activity on the PUSHPIN web application, store the user activity, and based on these, be able to recommend resources (e.g., users who bookmarked publication X also bookmarked publication Y; mutual followers; you might also assign these tags to the resource because others did so; people who visited this resource also visited that resource).

To sum up, the recommender system takes into account all the above information for recommendation. In addition, the recommendations will be textual and visual, and also can be explained to the user.

2.5 Visualizations

Visualizations prove very useful in presenting and understanding large and complex sets of data and mining for hidden patterns within them. They serve as a very useful decision support tool in research networks and help researchers to become and stay aware of large data sets [18,23]. Sometimes, they also allow interaction with the data in order to enhance the understanding [31]. In PUSHPIN, visualizations play an important part to support social awareness using a set of aesthetic visualizations of data related to researchers, affiliations and publications. We will have a brief look at some of the visualizations that we have or plan to have in PUSHPIN.

Usage and statistical visualizations This category of visualizations will be prevalent throughout PUSHPIN. For researchers, there will be a simple chart depicting the development of followers, co-authors, publications, etc. Similarly, there will be charts for a publication how the number of citations and bookmarks developed over time. Besides, visualizations based on general statistical data like typical co-authorship network sizes, most referenced articles, top research disciplines, etc. will have a place in PUSHPIN.

Trend-based visualizations This category will include trends using numbers as well as trends in usage of text over time. Trending citations, authors, topics and keywords will be visualized in appropriate manner.

Similarity-based visualizations Details of textual similarity between papers and bibliographic coupling similarity between papers will be explored here. Moreover, appropriate visualization of paragraphs that have been found during the near-copy detection will be developed and provided in PUSHPIN.

Map-based visualizations Geo-spatial visualizations show us the geographical location of researchers and institutions and help us understand the widely spread co-authorship networks and the associations of different institutions (inspired by the works of [17,18]). Particularly, we have interactive visualizations that show and link us to various information related to a researcher or an institution and relations between them.

Co-authorship visualizations For a researcher, there will be a circular visualization with the researcher at center and his co-authors around him in circles. This give us a chance to explore the co-authors of this researcher. When a user explores a discipline, a research interest, an institution or a tag, there can be sets of co-authorship networks related to the explore query which may not be connected. Hence, we do not use a radial layout here, instead build a graph comprising of different networks(not connected) to show various sets effectively.

Besides the above categories, we will also have tag-based visualizations like word clouds, spark lines, etc. and also circle-based visualizations

2.6 Email notifications

As Fred Wilson points out "*if you want to drive retention and repeat usage [of your service], there isn't a better way to do it than email*" [32]. Instead of making email disappear, social media has created new application fields for email and makes heavy use of them in all kind of domains. In PUSHPIN, we also use the power of email notifications to keep the users of the system up-to-date what is going on in PUSHPIN. Users will receive emails when they have new followers or someone comments on their publications. PUSHPIN will send alerts if it found new publications of an author or if someone tagged an author's publication. If users do not want to be bothered with emails, they can deactivate them or set adjust their granularity and frequency levels.

2.7 Access on mobile devices and interactive displays

In our previous research we found that mobile access to research information, together with context-awareness and push notification of relevant information is very relevant for researchers overall awareness of their research networks [25,23]. Moreover, research conducted by Nagel et al. [17,18] and Vandeputte et al. [28] shows that interactive tabletop applications are useful for sensemaking of publication data and co-authorship networks. Moreover, most of the existing social networks and Research 2.0 applications make allowance for the immense pervasion of mobile devices among all social classes by providing dedicated mobile applications the resemble the features of their web-based counterparts. Often, the mobile applications even make extensive use of the specific technical characteristics of the mobile devices such as camera, microphone, GPS positioning. Against this background, we decided to provide a mobile application, which could be used by all PUSHPIN users and a multitouch application that should be used for special occasions such as conferences.

The PUSHPIN$_{mobile}$ application resembles a significant part of the features of the web application. Making use of specific mobile interface patterns such as dashboards and multitouch gestures, researchers are enabled to access all the information from the social layer and to engage in social interactions with their peers. Researchers are also able to view their own and other researchers' profiles, search nearby researchers depending on their physical location, and also explore the different research disciplines, institutions and publications in the system. Moreover, researcher will also be able to tag other researchers and communicate with each other through private messages.

Beyond that, users of the mobile application will be enabled to authenticate and exchange data with the multitouch table application (PUSHPIN$_{MT}$). Therefore, researchers can connect to PUSHPIN$_{MT}$ using either Bluetooth or NFC. Additionally, the mobile application can bring up QR codes that can be scanned by the multitouch application. The QR codes can contain information about the researcher' own or other researchers' profile, institutions or publications. On PUSHPIN$_{MT}$, users will be able to explore their relations to other researchers and publications based on several scientometric measures. Moreover, they can explore the publications in PUSHPIN based on tags and other classifications. Finally, they can scan QR codes of any PUSHPIN object and get a virtual representation of the object on the tabletop.

3 PUSHPIN's eResearch infrastructure implementation

In this section we describe the technological underpinning of PUSHPIN's eResearch and big data analysis infrastructure and relevant technologies we employ in the realization of the PUSHPIN user interfaces.

3.1 Big Data analysis

In modern web-based (social) applications, users create huge amounts of data. This data can be used for analyzing the system or for building recommender systems to advance the user experience. For PUSHPIN, large computational power is needed to analyze uploaded scientific papers, do text extraction and manipulation, thumbnail creation as well as text analysis and similarity analyses, near-copy detection and metadata extraction. Most of the applied algorithms need large computational power and create huge intermediary data. To handle these needs, we decided to use well-known and massively scalable frameworks like Apache Hadoop[11] and Twitter Storm[12] for batch processing, handling large datasets and for real-time analysis. Both frameworks are designed for running on clusters of consumer PCs, are robust against system faults and optimized for highly parallel computation.

Storm is a distributed realtime computation framework developed by Nathan Marz. It consists of a master server called nathan, which controls a set of worker nodes called supervisors. The system is coordinated using the Apache Zookeeper framework[13]. A processing chain in Storm is described by a topology of steps called bolt and is filled with data by a datasource called spout. The spout and the bolts are distributed on a cluster of computing devices and connected to each other via messaging queues described within the topology. Each element of the topology will be created with a specific parallelism factor, which generates multiple instances of this element on different nodes of the cluster. The framework passes a computing object from the spout to the first bolt and then from bolt to bolt where several different tasks can be executed.

In PUSHPIN, Storm is used to do the first computing steps for an uploaded paper where near real-time responses are required. For this, we use multiple Storm topologies. If an user has uploaded a paper, the first topology receives the paper and extracts information, which are needed for rendering the next webpage directly after uploading the paper. After that, we can continue to asynchronously process the paper in order to extract information that takes more time to compute, like creating thumbnails of the pages, or doing text-processing, or update trend-detection values.

The Apache Hadoop framework consists of two modules which deal with the batch processing of big data: 1) the Hadoop Distributed File System (HDFS) and 2) MapReduce.

[11] http://hadoop.apache.org

[12] http://storm-project.net/

[13] http://zookeeper.apache.org

The Hadoop Distributed File System is an open source implementation of a fault tolerant, self-healing, distributed filesystem for large datasets inspired by the Google filesystem (GFS)[10]. It is designed to store large file, which are split and distributed over several nodes of a cluster, and to achieve high performance, while serving the data to computing processes. The processing methodology of Hadoop is an implementation of the MapReduce paradigm [5], which is designed to handle large amounts of data by splitting the input stream into chunks, which are computed on several nodes of a cluster. The MapReduce paradigm divides the processing into two stages to reduce the complexity. The first stage (map) processes several input key/value pairs and outputs a set of intermediate key/value pairs, which are sorted and transferred to the second stage (reduce). The reducer, eventually, merges all intermediate values, which are associated to the same key and outputs results for that key.

Hadoop provides batch processing function, which perfectly scales with the number of nodes in a cluster. This functionality excellently supports parallelism to a wide range of algorithms especially in data mining and information retrieval.

In PUSHPIN, Hadoop is used for several algorithms, which need large computational performance and that process big data. Amongst others, these algorithms compute the similarity of texts, clusters the papers, builds recommender models or run near-copy detection algorithms. Moreover, we use Apache Mahout[14] for the calculation of text-based similarities, text clustering, classification and recommender algorithms based on Hadoop MapReduce.

3.2 Text preprocessing

As described in Section 2.3, we perform several text preprocessing steps before a paper can be analyzed by the near-copy detection algorithm. The text extraction and thumbnail generation is done using Apache PDFBox[15]. Since many algorithms need to have knowledge about the language of a text, we use a Java-based language detection library[16] for that. The Part-of-speech tagging is realized using Apache OpenNLP[17]. Stemming and lemmatization of the extracted texts is implemented on top of the Mate Tools natural language analysis toolkit[18]. Finally, we make use of Apache Lucene[19] in the process of removing numbers and stop words that we consider as being not relevant for text similarities or near-copy detection.

3.3 Metadata and reference extraction

During the metadata and reference extraction processes we are trying to accurately detect a publication's title, author(s), contact information, like emails and

[14] `http://mahout.apache.org`
[15] `http://pdfbox.apache.org`
[16] `http://code.google.com/p/language-detection`
[17] `http://opennlp.apache.org`
[18] `http://code.google.com/p/mate-tools`
[19] `http://lucene.apache.org`

address data as well as author-provided keywords and the publication's abstract. Moreover, we are interested in the list of references and all the relevant data from each of the references. This metadata is extracted for different purposes, e.g., the attribution of publications to PUSHPIN users, the creation of co-authorship graphs, the calculation of recommendations and for detecting reference and research trends.

Once a publication has been uploaded to PUSHPIN and inserted into HBase, the metadata and reference extraction modules get triggered by Storm. The process involves triggering ParsCit and GROBID in parallel threads. GROBID (GeneRatiOn of BIbliographic Data) employs the concept of Conditional Random Fields (CRFs) for pattern recognition and data extraction [30]. Using this, "*GROBID extracts the bibliographical data corresponding to the header information (title, authors, abstract, etc.) and to each reference (title, authors, journal title, issue, number, etc.). The references are associated to their respective citation contexts*" [13]. ParsCit also employs the use of CRF model at its core for metadata extraction by *locating reference strings, parsing them and retrieving their citation contexts. It employs state-of-the-art machine learning models to achieve its high accuracy in reference string segmentation, and heuristic rules to locate and delimit the reference strings and to locate citation contexts.* [4].

Each tool does an independent metadata and reference extraction and the two results, obtained at the end, are then combined with potentially available other metadata like BibTeX data or PLoS XMLs. This merging is necessary as sometimes the metadata extracted from both tools differs, and also at times either of the tool misses out on some important metadata. If available, the data available in BibTeX or PLoS XML format are the most accurate source of information since they have been manually created by people knowledgeable of the publication.

3.4 Sign-up and sign-in using OAuth

In PUSHPIN, we use the Open Authorization (OAuth) protocol[20] to allow users to login to PUSHPIN using their Facebook, Twitter or Mendeley accounts. OAuth "*is a security protocol that enables users to grant third-party access to their web resources without sharing their passwords*" [11]. Apart from this, PUSHPIN also serves as an OAuth service provider, which implies that websites can use PUSHPIN for the sign-up and sign-in of users. OAuth is also used to connect the three PUSHPIN user interfaces to the backend.

3.5 The PUSHPIN API

In PUSHPIN, we use provide a REST (REpresentational State Transfer) API (Application Programming Interface) to communicate between the frontends (web-based application, mobile application and multitouch table) and the Java backend. The frontend sends/requests data to the backend using the REST API,

[20] `http://oauth.net`

e.g., information about a certain resource such as a publication. The backend in turn returns a representation of the resource in JSON notation. The reasons for using REST (over other available web services such as SOAP) are that it is light-weight, simple, very popular among web applications and that it provides better performance and scalability.

3.6 PUSHPIN user interfaces

PUSHPIN currently provides three user interfaces for its users. The web-based application serves as the main interface to our service and will be used by the average user. Moreover, we provide a mobile application for Android smartphones that allows the anytime-anywhere access to PUSHPIN's main features. Finally, we also provide a multitouch application for tabletop-displays that supports users in exploring the PUSHPIN data in new ways.

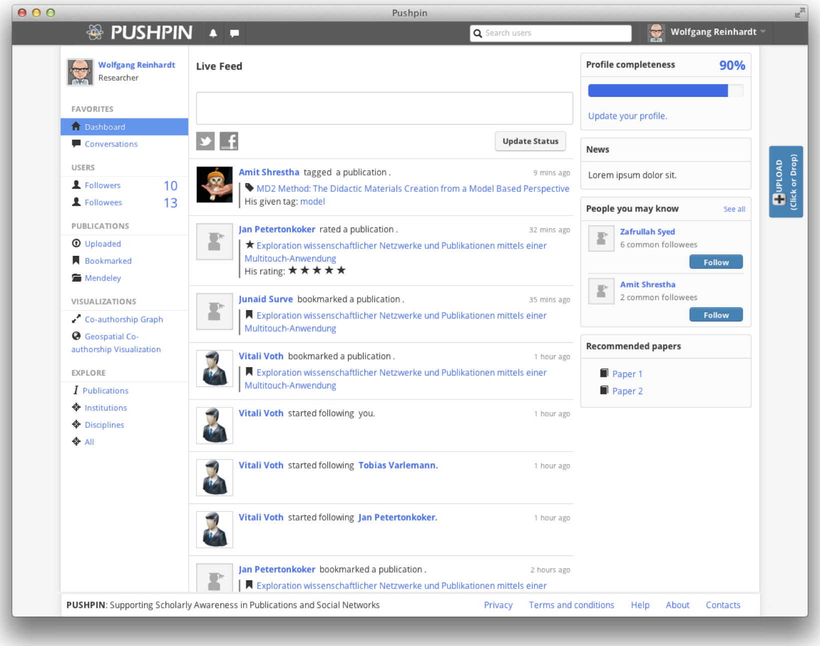
Web-based application The web-based PUSHPIN front-end is a self-contained application and serves as the primary application to most of the users (see Figure 1). This application is written in PHP5 and builds on the state-of-the-art in HTML5 and CSS3 development. It also involves extensive use of JavaScript that enhances the user experience. Also, various Javascript frameworks are used for different visualizations.

Mobile application The PUSHPIN$_{mobile}$ application is developed using the Android 4 SDK and supports all smartphones running Android OS 4.0 and higher. PUSHPIN$_{mobile}$ currently provides users an interface to the social layer of PUSHPIN and lets them flip through their activity stream, like and comment entries and post new status updates. The application can scan QR codes of any PUSHPIN object and present the data related to that object. Moreover, the users can locate themselves and see relevant researchers around them.

Multitouch application The main purpose of the PUSHPIN$_{MT}$ application is to provide different interactions with the data in PUSHPIN. In [24] we discern four basic modes of data exploration on PUSHPIN$_{MT}$: the 1) people-based, 2) topic-based, 3) event-based and 4) trend-based approach. Users can use the search to bring up researcher or publication profiles or authenticate themselves using PUSHPIN$_{mobile}$ or QR codes. Moreover, they can explore the relations between publications, which can be related by common references or authors, textual similarity or even by copied/cited paragraphs. Finally, users can explore the trends in reference and publication data as well as exploring the authorship patterns found during the automatic analysis of the publications.

4 Conclusion and future research opportunities

In this paper we have introduced the PUSHPIN approach for awareness support in research networks. In PUSHPIN we combine the best of two worlds:



Amit Shrestha tagged a publication . 9 mins ago
MD2 Method: The Didactic Materials Creation from a Model Based Perspective
His given tag: model

Jan Petertonkoker rated a publication . 32 mins ago
★ Exploration wissenschaftlicher Netzwerke und Publikationen mittels einer Multitouch-Anwendung
His rating: ★ ★ ★ ★ ★

Junaid Surve bookmarked a publication . 35 mins ago
Exploration wissenschaftlicher Netzwerke und Publikationen mittels einer Multitouch-Anwendung

Vitali Voth bookmarked a publication . 1 hour ago
Exploration wissenschaftlicher Netzwerke und Publikationen mittels einer Multitouch-Anwendung

Vitali Voth started following you. 1 hour ago

Vitali Voth started following Tobias Varlemann. 1 hour ago

Vitali Voth started following Jan Petertonkoker. 1 hour ago

Jan Petertonkoker bookmarked a publication . 2 hours ago
Exploration wissenschaftlicher Netzwerke und Publikationen mittels einer

Lorem ipsum dolor sit.

Zafrullah Syed
6 common followees

Amit Shrestha
2 common followees

PUSHPIN: Supporting Scholarly Awareness in Publications and Social Networks

Fig. 1. Dashboard in the web-based PUSHPIN application

classic features of Facebook-like social networking sites and those of innovative eResearch infrastructures. The integration of these features results in enhanced awareness support for researchers on both a social and a content layer. The recommender systems in PUSHPIN will not only recommend publications based on collaborative filtering but also on the actual content and reference data within the publications. Thus, PUSHPIN goes beyond the state-of-the-art and might help overcoming unwanted fragmentation in research networks and connecting researchers that otherwise would have stayed unknown to each other. In the coming months we will continue to improve the implementation of the analytical backend and further enhance the three user interfaces. We will invite selected users to an alpha test of the PUSHPIN web-based application in August and evaluate the existing features with them. The feedback on early versions of the software will help shaping the further development. We plan to release the system to public beta in early October 2012.

References

1. Salha Alzahrani and Naomie Salim. Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection. Lab report, Taif University Saudi Arabia and

Universiti Teknologi Malaysia, 2010.

2. Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 117–171. Springer Berlin / Heidelberg, 2002.
3. Simone Braun, Christine Kunzmann, and Andreas Schmidt. *People Tagging & Ontology Maturing: Towards Collaborative Competence Management*, pages 133–154. Springer, 2010.
4. Isaac G. Councill, C. Lee Giles, and Min yen Kan. Parscit: An open-source crf reference string parsing package. In *INTERNATIONAL LANGUAGE RESOURCES AND EVALUATION*. European Language Resources Association, 2008.
5. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.
6. Jyri Engeström. Why some social network services work and others don't — or: the case for object-centered sociality. Available online `http://bit.ly/eJA7OQ` (accessed 31 December 2010), April 2005.
7. Stephen Farrell, Tessa Lau, Stefan Nusser, Eric Wilcox, and Michael Muller. Socially augmenting employee profiles with people-tagging. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 91–100, New York, NY, USA, 2007. ACM.
8. Christiane Fellbaum. Wordnet. In Roberto Poli, Michael Healy, and Achilles Kameas, editors, *Theory and Applications of Ontology: Computer Applications*, pages 231–243. Springer Netherlands, 2010.
9. Christian Fuchs. *Handbook of Research on Web 2.0, 3.0, and X.0: Technologies, Business, and Social Applications*, volume II, chapter Social Software and Web 2.0: Their Sociological Foundations and Implications, pages 764–789. IGI-Global, Hershey, PA, 2010.
10. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, October 2003.
11. Eran Hammer. Introducing oauth 2.0, May 2010.
12. K. Knorr Cetina. Sociality with Objects: Social Relations in Postsocial Knowledge Societies. *Theory Culture Society*, 14(4):1–30, 1997.
13. Patrice Lopez. Grobid: combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the 13th European conference on Research and advanced technology for digital libraries*, ECDL'09, pages 473–474, Berlin, Heidelberg, 2009. Springer-Verlag.
14. Tamara M. McMahon, James E. Powell, Matthew Hopkins, Daniel A. Alcazar, Laniece E. Miller, Linn Collins, and Ketan K. Mane. Social awareness tools for science research. *D-Lib Magazine*, 18(3/4), 2012.
15. David R. Millen, Jonathan Feinberg, and Bernard Kerr. Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 111–120, New York, NY, USA, 2006. ACM.
16. George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.
17. Till Nagel and Erik Duval. Muse: Visualizing the origins and connections of institutions on co-authorship of publications. In *Proceedings of the Science 2.0 for Technology Enhanced Learning Workshop*, 2010.
18. Till Nagel, Erik Duval, and Frank Heidmann. Visualizing geospatial co-authorship data on a multitouch tabletop. In *Proceedings of the 11th international conference on Smart graphics*, SG'11, pages 134–137, Berlin, Heidelberg, 2011. Springer-Verlag.

19. Peyman Nasirifard, Sheila Kinsella, Krystian Samp, and Stefan Decker. Social people-tagging vs. social bookmark-tagging. In Philipp Cimiano and H. Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 150–162. Springer Berlin / Heidelberg, 2010.
20. Tim O'Reilly. What is web 2.0. Available online `http://oreilly.com/web2/archive/what-is-web-20.html`, 2005.
21. Tim O'Reilly and John Battelle. Web Squared: Web 2.0 Five Years On. Whitepaper, O'Reilly Media Inc., 2009.
22. Rob Procter, Robin Williams, James Stewart, Meik Poschen, Helene Snee, Alex Voss, and Marzieh Asgari-Targhi. Adoption and use of Web 2.0 in scholarly communications. *Phil. Trans. R. Soc. A*, 368:4039–4056, 2010.
23. Wolfgang Reinhardt. *Awareness Support for Knowledge Workers in Research Networks. Available online at http://bit.ly/PhD-Reinhardt.* PhD thesis, Open University of the Netherlands, 2012.
24. Wolfgang Reinhardt, Muneeb I. Ahmad, Pranav Kadam, Ksenia Kharadzhieva, Jan Petertonkoker, Amit Shrestha, Pragati Sureka, Junaid Surve, Kaleem Ullah, Tobias Varlemann, and Vitali Voth. Exploration wissenschaftlicher Netzwerke und Publikationen mittels einer Multitouch-Anwendung [Exploration of Research Networks and Publications using a Multitouch Application]. In Florian Klompmaker, Karten Nebe, and Nils Jeners, editors, *Proceedings of the 3rd Workshop Kollaboratives Arbeiten an interaktiven Displays [Collaborative Work on interactive displays] at the Mensch & Computer Konferenz 2012*, September 2012.
25. Wolfgang Reinhardt, Christian Mletzko, Hendrik Drachsler, and Peter B. Sloep. Design and evaluation of a widget-based dashboard for awareness support in Research Networks. *Interactive Learning Environments*, 2012.
26. Wolfgang Reinhardt, Christian Mletzko, Benedikt Schmidt, Johannes Magenheim, and Tobias Schauerte. Knowledge Processing and Contextualisation by Automatical Metadata Extraction and Semantic Analysis. In Pierre Dillenbourg and Marcus Specht, editors, *Proceedings of the 3rd European Conference on Technology Enhanced Learning (EC-TEL 2008), Maastricht, The Netherlands,*, volume 5192 of *Lecture Notes in Computer Science*, pages 378–383. Springer Berlin, 2008.
27. Ben Shneiderman. Science 2.0. *Science*, 319(5868):1349–1350, 2008.
28. Bram Vandeputte, Erik Duval, and Joris Klerkx. Interactive sensemaking in authorship networks. In *Proceedings of the 2011 ACM International Conference on Interactive Tabletops and Surfaces*, Kobe, Japan, 2011.
29. M.M. Waldrop. Science 2.0. *Scientific American*, 298(5):68–73, 2008.
30. Hanna M. Wallach. Conditional random fields: An introduction. CIS MS-CIS-04-21, University of Pennsylvania, 2004.
31. Matthew O. Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications.* Taylor & Francis, 2010.
32. Fred Wilson. Email: Social Media's Secret Weapon. Available online `http://articles.businessinsider.com/2011-05-15/tech/30100968_1_return-path-matt-blumberg-facebook`, May 2011.
33. Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.