© 2017 Aditi Adhikari
AUTOMATED PROFILE BUILDER: ONE STOP PORTAL FOR CREATING AND SHARING YOUR PERSONAL PROFILES

BY

ADITI ADHIKARI

THESIS
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:
Professor Kevin C. Chang
ABSTRACT

The explosive growth in digital presence have changed the way people form and stay connected to their professional networks, leading to the rise in multiple online profile management tools each catering to custom profile generation needs. With digital presence being split across multiple online tools, the challenge to maintain, update and customize details of all online profiles has been a challenge, perhaps more so in the academic researcher and professional community where a long list of publication and multiple pages of online profile is the norm. Today a typical researcher can have profiles that range from 1-2 pages like the typical industry standard to several pages as per typical academic standards where publications, patents, talks, workshops, etc are all part of a researchers profile. Maintaining this profile online across various independently run profile management system has created the problem of tedious multiple locations of data managements. As no existing profile management system leverages data from other systems it has been difficult for researchers to maintain and update online profiles. In addition, most of the profile management system have a fixed schema while creating online profile so the user does not have the flexibility to create a customized profile. In this thesis, we propose an Automated Profile Builder (AutoPB) which acts as a one stop portal for all the online profile management system. AutoPB aims to solve the issues surrounding current distributed and scattered profile management systems by its three main features: a) schema sharing, b) centralized multi format data in/out portal and c) client-side data retrieval system. First concept is known as schema sharing, which is a unique feature that no other profile management system has. This feature lets user to learn and share schema that are used while building an online profile. Secondly, the system provides flexible Import and Export feature. The user can import data from external sources as well as export the profile in different formats. Lastly, the profile builder performs client-side search-
ing, crawling and parsing to minimize the server work load. Using our profile builder users can import data from their existing online profile and would not need to manually type their information which is easily available in the internet. In this thesis, we demonstrate that AutoPB with these features can act as a one stop portal especially for researchers to more easily manage their online profile by centralized data management paired with distribution of multiple custom profiles.
To my parents and my sisters, for their love and support.
I would like to express my gratitude to Professor Kevin C. Chang for being a wonderful mentor and not just an advisor. Kevin has always been available for advice and would always guide me to think out of the box. He has helped me tremendously to choose my thesis topic. I have gained a lot from my interactions and time with Kevin his dedication and commitment to work has been inspiring and will stay with me for a long time.

I would also like to thank all the members of FORWARD group and all the DAIS professors from whom I have learned a lot.

My heartfelt gratitude to my parents and family for encouraging me to aim higher. Without their love and support this would have not been possible. My two sisters- Anku and Anju who have always encouraged me to dream big and achieve bigger. Their support and encouragement over the years has been the greatest gift I have received. Lastly, I would like to thank all my friends, who always managed to bring joy in the time of stress and for always being there for me.
# TABLE OF CONTENTS

LIST OF TABLES ................................................... vii

LIST OF FIGURES ................................................... viii

CHAPTER 1 INTRODUCTION ........................................... 1
  1.1 Motivation ................................................... 2
  1.2 Functionality ................................................. 3

CHAPTER 2 RELATED WORK ........................................... 5
  2.1 Research networking tools ..................................... 5
  2.2 Vertical search engine for People ............................. 10

CHAPTER 3 PROFILE BUILDER ...................................... 12
  3.1 System architecture ........................................... 13
  3.2 System functionality .......................................... 14

CHAPTER 4 SCHEMA SHARING ...................................... 20
  4.1 Implementation ................................................. 22
  4.2 Workflow for schema sharing .................................. 23

CHAPTER 5 EXPORT AND IMPORT ................................ 25
  5.1 Import module .................................................. 26
  5.2 Export module .................................................. 35

CHAPTER 6 CLIENT-SIDE SEARCHING, CRAWLING AND
  PARSING .......................................................... 36
  6.1 Google Chrome extension for LinkedIn ....................... 38

CHAPTER 7 SYSTEM DEMO ............................................ 40

CHAPTER 8 CONCLUSION AND FUTURE WORK ....................... 47

REFERENCES ........................................................ 48
LIST OF TABLES

2.1 Comparison of existing research networking tools with our system ........................................... 9
# LIST OF FIGURES

2.1 Result from Pipl when searching for Jiawei Han .......................... 11
3.1 System architecture of the Automated Profile Builder ................. 13
3.2 Schema sharing functionality .............................................. 15
3.3 Add new attribute functionality ......................................... 15
3.4 Add, edit and delete an attribute ........................................ 16
3.5 Attribute selection functionality ......................................... 17
3.6 Sort, re-order, filter and select data functionality ...................... 18
4.1 List of schemas for publication ............................................ 20
4.2 Workflow for schema sharing module ..................................... 24
5.1 External sources integrated in AutoPB .................................... 26
5.2 Google Scholar profile for Professor Jiawei Han ....................... 27
5.3 Google Scholar publication imported in AutoPB ......................... 28
5.4 JSON output from the NSF API .............................................. 29
5.5 Awards from NSF imported in AutoPB .................................... 29
5.6 Patents returned from search in Justia website .......................... 30
5.7 Patents from Justia imported in AutoPB ................................... 30
5.8 Publication from Science direct imported in AutoPB .................... 31
5.9 Education and Employment shown in the LinkedIn profile .......... 33
5.10 Education and Employment imported from LinkedIn ................. 33
5.11 Video lectures returned by the search result in Videolectures.net website .................................................. 34
5.12 Video lectures imported from Videolectures.net in AutoPB ......... 34
5.13 API output for Jiawei Han showing Honors ............................. 35
7.1 Homepage of AutoPB where user inputs name and affiliation ........ 40
7.2 AutoPB when a new user uses the system ................................ 40
7.3 User Interface to search for existing attributes along with their schemas .......................................................... 41
7.4 User Interface to add new attribute or existing attribute with new schema ....................................................... 41
7.5 User Interface to add attribute from LinkedIn .......................... 42
7.6 Shows the Google Plugin in the top right corner of the window. When the user clicks the icon the data is fetched from the current LinkedIn profile page ................. 42
7.7 User Interface after adding and importing all the attributes . 43
7.8 User Interface to edit records, add or delete fields .......... 43
7.9 User Interface to select attributes for the user’s profile .... 44
7.10 User Interface to select data .................................. 44
7.11 User Interface showing paginated data sorted by PaperYear . 45
7.12 User Interface showing template selection where the user can select from existing templates .................. 45
7.13 Output resume in PDF format ............................... 46
CHAPTER 1

INTRODUCTION

With the explosion of connections in the research community and the availability of more and more online profile management tools, researchers are the main group of users, facing the challenge of updating and maintaining online profiles in the form of short resume and multi-paged detailed academic style CVs and updating their online presence timely. People no longer need to rely upon a personal connection to support and spread their professional activities as everything is on the internet. But maintaining online profiles can be quiet tedious.

Search engines today have facilitated finding people on the web and reading their profiles and details shares. This is such a common practice that according to Bing, people searches account for about 10 percent of all search on Bing[1]. However, it can be very time consuming to search for such information manually as the information of interest is scattered across many different web pages. With the proliferation of various academic profile management system its hard to get all the information of a person from one single source. So people generally browse through all the web links that a single search engine returns in order to gather the information about a person. In this project, I have developed an automated profile builder which integrates data from external sources. The main objective of the project is to create a platform in which the users can import their information from all the external sources, edit it and export it in different forms. The system aims to be a one stop portal for researchers to create their profile also combining information they are already maintaining in different places in the web, if applicable.

While creating accounts on platforms like ResearchGate, LinkedIn etc., the users need to manually type all their information which can be very tedious and repetitive. Apart from the users who are creating an account, users who are hunting for profiles or information about other researchers also have to do tedious searches across multiple profile management systems, which may not
all be updated and maintained equally. In this system we try to automate this process and provide the user a functionality to import their information from different external sources.

With the increase in the number of profile management systems, researchers end up creating profiles in multiple platforms. There is no profile management system that leverages data that already exists in other online profiles. This makes the process of maintaining and updating online profile time consuming and laborious. In addition, existing profile management system use fixed schema to define each attributes of the online profile and does not provide the user flexibility to customize their profiles. These are the issues that we addressed while designing our automated profile builder. In our system, user can import their records from other external sources making use of the information that they already have online. By leveraging information of the user that is already in other online profile management system the user does not have to type information manually if it is already available in the internet. The system also provides the functionality to choose schema for each attribute and customize it. The users can select attributes for each of their online profiles and customize the details which makes it easier to use this system to create profiles with different requirements. Apart from choosing their own schema the user is also able to access profile attributes created by other users, allowing the platform to grow through contribution and adoption of the users themselves.

1.1 Motivation

According to a survey conducted by ORCID in 2012 [2] only 50 percent of researchers have an online profile and only a quarter of researchers who work for more than one institution have a completed record. In the survey, they found that three quarters of researchers used a word processing program to maintain their profiles as opposed to a database or web services. Of the other one-quarter 9 percent used a website and 21 percent used another type of software.

When professors and researchers apply for grants, apply to a talk at a conference they need to send their resume. Different grants or conferences have different resume format that they prefer. So, the researcher needs to
manually edit their resume to satisfy the requirements which is very time consuming. Another scenario is when a researcher wants to add something to their profile for example, if he/she publishes a paper. In this case, the user goes to his/her homepage and all other academic profile websites to add his published papers. Amending and adding data to your resume or online profile is done very frequently. So to amend it in every profile that you have created can be very laborious.

This motivated us to build this AutoPB which automates the process of building a profile and provide users with user friendly interface for to edit, import attributes of their profile easily and export it. Even though the major group that we are focusing on this projects are researchers, the system can be used by any user.

1.2 Functionality

The main functionalities that our system provides are as follows:

- **Schema sharing:** The system provides a function in which users of the system can browse and learn what attributes other users used to build their profile along with its schema. Apart from learning from other users schema the user has flexibility to edit, add or remove any fields in an attribute.

- **Flexible profile import and export:** The system provides option to export the profile as an online profile or PDF resumes. Additionally, we also developed an API which the user can use in their website to import data. The system also supports import from other external sources like Google Scholar, Justia, DBLP, LinkedIn, VideoLectures.net, NSF, Science Direct.

- **Client side search:** While fetching data from external sources, the crawling and parsing of the data is done on the client side and not in the server side. This provides an advantage over doing in the server side when the number of users in a system increases making it easier to scale up. The number of API request that we can send to external sources can be limited so to overcome this the search takes place in the client
side in our system. This was one of the main objective of our system and was very challenging because of browser and service limitations.

We will be discussing in more detail about each of the functionalities mentioned above in later chapters of this thesis. The outline of the rest of the paper is as follows:

- In Chapter 2, we discuss about some related work and discuss how our system is different from other existing system.

- The system design and system architecture of the Automated Profile Builder is described in Chapter 3.

- In Chapter 4, we describe the schema sharing functionality which is a unique functionality which no other system has currently.

- In Chapter 5, we describe how our system provides flexibility to export and import data from external sources.

- Then, in Chapter 6 we describe how we implemented client side search in our system. We will be discussing the advantages doing client side search compared to server-side.

- We will be showing screen-shots of the system we built in Chapter 7.

- Chapter 8 concludes the paper and discuss about the future works that we can incorporate into the system to make it better.
CHAPTER 2

RELATED WORK

Our system aims to build an automated profile builder integrated with existing profile management systems. In this chapter, we study about current academic profile management system and discuss how they differ from our system that we developed. Our system though focuses on creating profiles can also be used in a vertical search engine. So, we also discuss about a popular vertical search engine known as Pipl in this chapter.

2.1 Research networking tools

Research networking (RN) tools are used to discover and use research and scholarly information about people and resources. The tools serve as knowledge management systems for the research enterprise. RN tools facilitate the development of new collaborations and team science to address new or existing research challenges through the rapid discovery and recommendation of researchers, expertise, and resources[3][4].

RN tools are different from general search engines as they access information from other database and are not limited to web pages. They are also different from social networking site like LinkedIn, Facebook, etc., and more reliable as they only represent the data coming from verifiable sources and do not rely only on the information given by the users[5].

There are many RN tools available and developed by universities. But most of them are limited to use within universities or organization or they are part of a proprietary system. For example: VIVO is a popular RN tool. By populating the system with researcher interests, activities, and accomplishments by an institution, VIVO enables the discovery of research and scholarship across disciplines at that institution and beyond[6]. Similarly there are other tools like ArnetMiner [7] that provides the functionality to automatically ex-
tract information about a researcher from the web and discover connection between researchers. But this system does not provide the functionality to maintain researchers online profile nor does it provide a flexibility to import and export from other external sources.

In the next section, we will be focusing on summarizing existing academic profile management systems that are used nowadays. We will also be using some of these systems as a source that the users can import their data from in our system.

2.1.1 ORCID

The Open Researcher and Contributor ID (ORCID) is a system created to solve the researcher name ambiguity problem. The core mission of ORCID is to provide a registry of persistent unique identifiers for researchers and scholars[8]. Apart from researchers, publishers are also affected by the name ambiguity problem and invest substantial work to disambiguate and manage. Incorporating unique author identifiers provided by ORCID is a way to solve this issue but to be effective it needs to be incorporated in publisher workflow. So, the ultimate goal for ORCID is to have this unique identifiers used whenever a scholarly contribution is made. The ORCID identifiers then can facilitate the creation and maintenance of author profiles and support importing information from other external sources. ORCID has a collaboration with number of publishers and service providers like Springer, Wiley, Association for Computing and Machinery, Scholar One, etc. Their system support importing user’s information from a number of external sources.

ORCID also offers a public API that allows other systems and applications to connect to the ORCID registry. The API allows i) users to sign into your system using ORCID username and password ii) gets a user’s authenticated ORCID id, and iii) perform a machine-generated search of the ORCID registry. While creating a profile in ORCID, even though the publications are identified based on user’s information, all other basic information needs to be entered manually. The system has a fixed schema for each attribute so the user is not able to customize the information that is shown in the profile. On the other hand, our profile builder is designed so that the user can choose attributes and the fields of each attributes as well as the order of fields and
make their own customized profile.

2.1.2 Google Scholar

Google Scholar is a tool used widely for searching scholarly information. It facilitates searching of conference papers, articles, books and other scholarly articles in any field. The crawler of Google Scholar crawls huge database of well known scholarly publishers and university presses (such as IEEE, ACM, Macmillan, Willey, University of Chicago) and other digital hosts, scholarly organization and government organizations[9]. While searching in Google Scholar it performs multiple database search by default and returns bibliographic records, abstract and snippets of the context from the full text matching the query. Apart from this, you can also search for researchers profile with their publications along with citation, total citations, h-index, current affiliation, email address, research interest.

While creating an account in Google Scholar the attributes of the profile is manually entered by the user. The publications for the user is automatically listed with an option to add it to the user’s profile. Google Scholar does not offer any API or SDK. So the only option we have is to manually crawl and parse the web page to get the relevant data out of the page.

Google Scholar is the most popular tool used to search publications, articles or any other scholarly information. Users can create profiles in Google Scholar but solely it cannot act as an online profile as it is missing information such as Education, Employment, Awards etc. While designing our system we built in a way that the user can add all the necessary information that they want so it is not limited to only publication or scholarly articles.

2.1.3 Academia.edu

Academia.edu is a social network designed specifically to facilitate those associated with academic institutions and specializing in academic activities like sharing papers and data sets. The system tracks various metrics showing users how many times their profile was viewed, how many times documents have been viewed and even the users that led people to their profile.
While creating a profile in Academia.edu\footnote{https://www.academia.edu/} I noticed that I had to manually enter all the information including my publications. It does not support importing from existing external database or any bib files. Therefore, it was mainly designed to publicly share work by uploading full papers manually. But the process of manually uploading papers is very time consuming and tedious especially for researchers who have 100s of papers. To overcome this problem and to make the system easy to use when building our system we included a functionality that lets user to import from external sources. Similarly, like other tools Academia.edu also has a fixed schema defined for each attribute.

2.1.4 ResearchGate

ResearchGate is another academic network used by researchers. Apart from researchers profile, it has a question and answer feature which aims to crowd source problems and get help from researchers in the same research area.

ResearchGate has its own measurement called RG Score, that assigns members a score based upon content interactions and the score of the members interacting with the content. This score is also influenced by content contributed to ResearchGate, like profile information and answered or asked questions, in addition to publication information, like views, downloads and citations\cite{10}. ResearchGate maintains its own database that the user can search in order to add publications but the database is based on other users entry. So, if your co-authors have added a paper with you it will be able to find it otherwise you need to manually enter all the publication information.

2.1.5 LinkedIn

LinkedIn is a professional social network mainly used for professional communication and career networking. It aims to connect people via work relationships, especially through indirect connections \cite{11}. LinkedIn does not contain academic-specific features like reference managements and is not designed for academic networking because disciplines are already well organized through conferences, web sites, and academic publications \cite{11}. Though we
can add publications and patents in LinkedIn, nowadays all of the data needs to be manually entered which can be troublesome. Similar to other system, LinkedIn also has a fixed schema definition for each attribute.

<table>
<thead>
<tr>
<th>System</th>
<th>Data Entry</th>
<th>Schema</th>
<th>Import from other sources</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCID</td>
<td>Manual entry of basic information</td>
<td>Fixed schema with fixed attributes</td>
<td>Available from member organisations</td>
<td>Profile with Education, Employment, Works, Funding</td>
</tr>
<tr>
<td>Google Scholar</td>
<td>Manual entry of basic information</td>
<td>Fixed Schema with fixed set of attributes</td>
<td>Available for Publication</td>
<td>Profile with Picture, Research Interest, Current affiliation, Publications</td>
</tr>
<tr>
<td>Academia.edu</td>
<td>Manual entry of all information</td>
<td>Fixed Schema</td>
<td>No</td>
<td>Profile with Current Affiliation, Biography and Papers</td>
</tr>
<tr>
<td>ResearchGate</td>
<td>Manual entry</td>
<td>Fixed Schema</td>
<td>No</td>
<td>Profile with fixed attributes users added to the profile</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>Manual entry</td>
<td>Fixed Schema</td>
<td>No</td>
<td>Profile with fixed attributes users added to the profile</td>
</tr>
<tr>
<td>Our System</td>
<td>Manual and import both</td>
<td>No fixed schema</td>
<td>Available from different sources</td>
<td>Customized profiles with different attributes</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of existing research networking tools with our system

Table 2.1 shows the difference between existing systems and our system. One thing we noticed that no other system provide is the flexibility to define a schema for each attribute. Additionally, no systems leverage data from other existing profile management system and ends up working independently. We wanted to address this issue so that updating and maintaining online profile
is easier for the user. In our system, we provide the user platform to edit and create their own customized attributes. We also build an API so that integration with other system is easier. In addition, the final output or the profile from all the existing system is fixed you cannot generate different profile for different requirements. Most researchers use word processing tools to edit or update their resume because of this lack of flexibility to generate customized profile. There is no profile builder that assist in building profiles according to the user requirements. While building our system we address all of these issues by giving users the flexibility to create different profile each time and export it choosing from a list of templates designs.

2.2 Vertical search engine for People

Pipl is a well-known deep web crawler. Unlike other typical search engines, Pipl focuses on searching personal information. A user can search based on a person’s name, username, email address or even phone number [12]. Pipl performs most of its search in real-time, i.e. the engine actually queries its sources on the fly. Pipl crawls websites such as LinkedIn, Twitter, Facebook, Flickr, Pinterest, Amazon, YellowPages. Figure 2.1 shows the result from Pipl system 2. Pipl provides following information about people: Name, Gender, Country of origin, email address, Education, username, phone number, Address, image, Age. So Pipl is designed to get basic information about the people which is different from our system which is designed especially for researchers and can search for attributes like publications, honors, etc.

\footnote{Pipl: https://pipl.com/}
Figure 2.1: Result from Pipl when searching for Jiawei Han
CHAPTER 3

PROFILE BUILDER

In this thesis, we are building an Automated Profile Builder (AutoPB) and integrating it with existing profile management systems. Using our system the user will be able to import their data from other external sources so that they don’t have to start from scratch by manually typing all the information. Unlike other systems that we discussed in Chapter 2 we wanted the system to be more flexible so that the user can edit and generate customized profiles with different attributes and data. AutoPB aims to be a one stop portal for researchers to make their online profile or resume. It aims to reduce the burden for researchers to report and maintain their online profile.

AutoPB is integrated with other external sources so that the researchers don’t have to fill out information manually. Integrating other sources with our system makes our system more attractive to the users as they can import their existing data in other profile management system to generate a customized profile. This way, it can leverage the data from other external sources as well as the user has the flexibility to manually enter or create attributes. The Profile Builder that we are building consists of the following 3 main concepts which makes the system different from the existing Profile Builders that we discussed in Chapter 2:

- **Schema sharing**: Schema sharing is a unique concept that no other existing online profile system has. The user can select schema for each attribute based on what other users added and from external sources. It makes the system easy to use for new researchers or graduate student who would want to learn from other researchers on what to put on their online profile/resume.

- **Flexible import and export**: The system is built to be very flexible providing options to import data from other external sources. Apart from importing it also allows users to edit the imported data. The
edited attributes can be saved in the database so that the user can retrieve it later. Likewise, for the export module we have built an API that the user can call to retrieve their data in json format. Furthermore, they can create their online profile or select a template to create a resume and save it as a PDF file.

- Client side search: In the automated profile builder that we built we made the search module as well as the entity extraction module in the client side. This was the main challenging part of this project. As many existing systems run their algorithm on server, client side scripting is not commonly used. We designed our system in a way that all the process takes place on the client browser. The main advantage being that the load on server side will be reduced and the system can be easily scaled up. This gives us many advantages compared to doing it on the server side that we will discuss later in Chapter 7.

Figure 3.1: System architecture of the Automated Profile Builder

3.1 System architecture

The system architecture is shown in Figure 3.1. The input for the system will be the name and affiliation of the user. The Profile Builder checks with mongodb if the user is a new user or an existing user. If the user has past
data then it loads their data from the database. If the user is new or existing regardless of that the user can import attributes from external source or add a new attribute manually to the system. After importing, the user can also edit their records by deleting, adding or editing their imported records. After the data is edited, its saved to the database. The third step is the attribute selection step in which user selects the attribute that they want in their profile. In the fourth step, the user can organize the selected attribute records by sorting and selecting the records, filtering fields and ordering the fields. The fourth step is followed by template selection in which user selects from a list of templates to generate a PDF resume. Finally, after the system generates a profile or resume, the user can download the resume or share their online profile to others.

3.2 System functionality

The system is divided into different modules as seen in the Figure 3.1. I will be describing the functionalities that each module provides in our system:

1. Edit and Import module: This module provides the user flexibility to edit the data as well as import attributes from different external sources. The functionalities in this module are:

   - Schema selection: Users can select from a list of schema for each attribute to add it to their profile. The user can select from a list of schema based on external sources or based on what other users of the system have used to define an attribute. This functionality aims to make it easier for researcher to learn and share with their community when it comes to creating online profile. Figure 3.2 shows an example of the schema selection functionality. User can see a list of attributes created by other users as well as from other external sources from which they can choose from. For example: for the ScienceDirect-Publication there are two entries, for the first one the source is Science Direct so the data is imported from Science Direct and the second one the data is manually given by other user of the system.
Figure 3.2: Schema sharing functionality

- Adding new attribute: If a user does not find an attribute in the list, the user can add a new attribute to their profile whose schema is automatically added to the database in the system. When another user signs in to the system the user is able to see the new attribute added by the other users of the system. As shown in Figure 3.3 while adding a new attribute the user needs to input the name of the attribute, description and the fields that the attribute contains. All the attributes have a list of fields that describes the attribute and its corresponding data type.

Figure 3.3: Add new attribute functionality

- Adding new field to an attribute: When a user adds a new or existing attribute, the user also have an option to add a new field to that attribute. As shown in Figure 3.4 if the user wants to add...
a new field for example the Venue then the user needs to select
the type of the field and name of the field and simply click the
Insert Column button which would insert a new column to the
existing records. The user can also see a list of fields that other
users added in the drop down list.

• Deleting a field in an attribute: Different users have different re-
  quirements. Some users might want additional information in each
attribute, whereas some may want very less fields. In AutoPB
users can delete a field from an attribute and customized their at-
tribute. For example, in Figure 3.4 if the user does not want the
Date field she can simply click the cross sign next to it to delete
it from all the records.

• Adding a new record to an attribute: Users can add new record to
  any of the attribute. For example, in Figure 3.4 a user can add a
new record by click on the Add ScienceDirect-Publication button.

• Edit existing records in an attribute: Users can edit existing
  records imported from external sources.

Figure 3.4: Add, edit and delete an attribute

• Deleting a record from an attribute: User can delete a record if
  they don’t want it to be included in their profile. As shown in in
Figure 3.4 each record has an edit and Delete button on the right
most column.

• Importing from different external sources: This is one of the main
  functionality where users can import from other external sources.
Right now the system supports import of Publication, Profile Pic-
ture, Research Area, Home page, Awards, Patents, Location, Ed-
uation, Employment, Summary, Current Employment, Current
School, Video lectures from sources like Google Scholar, NSF, Science Direct, Justia, LinkedIn and Videolectures.net.

- Saving the edited profile: After the user is done with editing his data and attributes he can save it so that when the user logs in back to the system the latest saved records is shown. The user can edit it again and save it in the system.

### Select the attributes for your CV

- Patents
- ScienceDirect-Publication
- Videos
- NSF-Award
- GoogleScholar-Research-Interest
- GoogleScholar-Current-Position
- GoogleScholar-Publication
- Skills
- GoogleScholar-Picture
- Current-Position

**Next step**

Figure 3.5: Attribute selection functionality

2. Attribute selection: After users edit their records, the next step is to select the attributes that they want to include in their online profile/Resume. There can be different requirements when you are creating a resume. In some cases, some prefer 1 page resume or a 2 page resume or a resume that includes everything. So, we provide a feature in which the user can select what attributes they want in their profile that they are building each time they create an online profile/Resume. As shown in Figure 3.5, the user can select the attributes that they want each time they create a profile. This is a new concept that no other existing system has incorporated.

3. Sort, re-order and filter: After the attribute selection module, in this module the user can play around with the records and select the records
for each attribute they want to be included in their profile. The functionalities for this modules are described below:

Figure 3.6: Sort, re-order, filter and select data functionality

- Sort records: Users can sort records selecting any fields in an attribute. This makes it easier for user to select records based on any fields. For example: if a user wants to select their top 20 recent publication they can sort the record based on the published year and select it.

- Select records: After selecting the attributes in step 2, for each attribute the user can select the records that they want to include in their profile. A user might have 100s of publication as shown in Figure 3.6 where the user has 10 pages of publication but might only want few of the publication while creating a short profile. Our system provides an easy way to select the data from each attribute before generating a profile.

- Filter fields: Another functionality is for each attribute they can hide/unhide a field. If they don’t want to include a particular field for an attribute they can just hide it. For example: in the Figure 3.6 if the user want to hide the citation field they just need to click the Blue link under the GoogleScholar-Publication title. After hiding it the field won’t be used while generating the profile of the user.
• Re-order fields: This is another interesting functionality in which user can re-order the fields in each attribute. The order will donate the order in which it will be displayed in the generated profile. For example if the user want the PaperName before the Authors, the user just need to drag and drop the Authors after PaperName. Different users have different requirements for the order. So instead of fixing the order we make our system more flexible giving the user an option to reorder the fields for each attribute.

4. Template selection: In this module, the user can select from a list of templates to generate an online profile or PDF Resume.

5. Output: The last module includes the generated online profile or Resume as the output of the system. The user can save the Profile as a PDF or print it.

6. API: We have also created an API for the system that the user can use to export their profile. The API returns the output as JSON format. Using the API the user can easily integrate the data from AutoPB to their website or other places.
CHAPTER 4

SCHEMA SHARING

In this Chapter, we will be introducing a new concept called Schema sharing that we have implemented in our system. Before discussing the details we define some terms that we will use throughout the chapter:

- **Schema**: Database schema is the structure of the database that defines the objects in the database. Similarly, schema in AutoPB is the structure that defines the attributes. One attribute can have multiple schemas as shown in the Figure 4.1, Publication has different schemas that the user use. The first schema contains PaperName, Authors, Venue, Citation and Year whereas the second one contains PaperName, Authors, Venue and Year. Similarly, the third one contains an additional field URL.

- **Attribute**: Attribute in the online profile are aspects such as Publications, Education, Employment, Awards, Books, Contact Information etc.

- **Fields**: Fields are what defines an attribute. An attribute is defined by a set of fields. For example in Figure 5.1 PaperName, Authors, Venue, Citation, Year, URL all are fields that defines Publication. Each field is associated with its corresponding data type.
• Source: Each attribute has a source attribute. Attribute manually entered by the user will have Source as Manual. An attribute imported from other sources will have the source name as the source. For example: Publication can have different source: Google Scholar, Science Direct etc.

Schema sharing is a unique concept that we have implemented in AutoPB that no other profile management systems has. All the existing online profile management system that researchers use has a fixed schema for each attribute. Therefore, users are not able to add a field or delete a field from the schema if they want. Apart, from making the system flexible, schema sharing also induces an environment of learning from other users. For example: If you are a new researcher or a graduate student and you are applying for different positions in industries and academics. It is common for you to refer to other people’s resume to learn what to include in your resume. AutoPB makes this process easier by listing out the schemas other users of the system used for each attribute. The user has an option to choose from the list of existing schemas or add their own schema and use it. Researchers while creating their Resume/profile often include attributes or fields based on the requirements. Since most of the existing systems does not provide flexibility to build a customized profile users often use text editors in order to create profiles. The main aim, of implementing this unique concept of schema sharing is to provide users the flexibility to customize their profile according to their requirements.

Apart from other user defined schema, there are fixed schema imported from other external sources like Google Scholar, NSF, Justia, Science Direct, LinkedIn etc. If the user selects a schema which has an external source, they can import all their data as well as the schema used in the external source. The user can also edit the schema by adding or deleting the fields after importing.

We are using templates in our system to display different attributes in the profile. So while designing the schema sharing functionality the first method I tried was to create different templates according to the user inputs. This process resulted in creating template file each time a user added a new schema to the system. As the number of users increased in the system this would result in 1000s of schema being created resulting to 1000s of templates which
was not a good solution when the system scales up. So instead of creating a template each time the user adds a new schema I came up with the idea of storing the schema in the database as it is more cost effective in terms of storage rather than saving the whole template. The schema for each attribute is stored in the database with a list of fields and its corresponding data type. Whenever a user selects an existing schema, the schema is mapped to a dynamic template which creates the table to display it dynamically.

4.1 Implementation

Schema sharing is implemented in the system by syncing it with the database. We choose Mongodb as our database because being a non-relational database its easier to store data without a fixed schema. In our Mongodb we have 3 collections that we use:

- Resumes: We use this collection to save the resume for all the users. Each document consists of user name, attributes, fields and data type of the field along with all the records stored in it.

- New attribute: We use this collection to store the schema of all the attributes the user added in our system. Below are some examples on how each attribute is stored with a new schema. Each attribute has a set of fields with their data type, Name of the attribute, Description of the attribute (optional) and the Source of the attribute. Source can be external sources or if there is no source it will be listed as "Manual" that means the user needs to enter their information manually.

```json
{
  "_id" : ObjectId("58f12186db23a630f2002e90"),
  "attribute" : "Publication", "Description" : "This attribute is for Paper or Publication ", "Source" : "GoogleScholar", "fields" : [
    { "Name" : "PaperName", "type" : "text" },
    { "Name" : "Authors", "type" : "text" },
    { "Name" : "Venue", "type" : "text" },
    { "Name" : "Citation", "type" : "number" }
  ]
}
```
4.2 Workflow for schema sharing

The workflow of schema sharing is shown in Figure 4.2. The first step is all
the schemas stored in the database are fetched and grouped together based
on the name of the attribute. A list of schemas for each attribute is shown
to the user. The user has an option to select from the list or add a new attribute
or schema. When the user adds the new attribute, its automatically saved
in the database and the other users of AutoPB can access it. After step 2 of
adding the schema/attribute the user can edit the schema according to their

• New fields: We use this collection to store new fields that the user
added to each of the attribute. An example for Employment is shown
below. Fields is a list containing the Name and the type of each fields.

    { "id" : ObjectId("58db57b5c29e65e64e9b1105"),
      "attribute" : "Employment", "fields" : [ { "Name" : "Supervisor", "type" : "text" },
      { "Name" : "StartDate", "type" : "date" }, { "Name" : "StartYear", "type" : "year" } ] }


requirement- adding and deleting fields. After the user is done editing and adding all the records, the user saves it. This will result in all the edited schemas to be stored as new schema if it does not exist in the database. While storing in the database, we check whether the resulting edited schema is same as the schema in our database if so we won’t be adding it avoiding duplicate schemas to be stored.

Figure 4.2: Workflow for schema sharing module
CHAPTER 5
EXPORT AND IMPORT

Different users have different requirements. Most of the researchers will have at least one online profile that they already have their records stored on. The user with an online profile or records available would want to use those records and not type it again manually in the system. Using AutoPB user can import different attributes from different external sources. This functionality aims to reduce the burden of manually entering records which can be very tedious.

The main motivation came while using existing systems like ResearchGate and Academia.edu. Manually typing information about publications can be very time consuming especially if you have 100s of publications that is easily available in GoogleScholar. Instead of typing out all the information from scratch the users would want to use already available information from external sources. This would also help the users to create a framework to automatically update their profile based on the changes in the external source. For example, if you have a new publication and its added to google scholar, you would want a system that automatically fetches this added information so that you don’t have to manually address this. Our system provides this functionality reducing the effort for researchers to maintain and create their profiles.

Importing from other external sources can be very tricky. Some sources provide public API that the users can use to fetch the information whereas tools like Google Scholar does not provide any API or SDK. In cases like this, we tried crawling the site directly but getting the web page with the target entity can be difficult. For example, if you are crawling for a researcher name Kevin Chang there might be more than one Kevin Chang that the Google Scholar returns. In this case, we use some guidance from users by asking them the Google scholar url that they want to import their data from. In addition, Google Scholar contains many information like Picture, Current
Affiliation, Research Interest, Publications. Users might only want to import the Publications from Google Scholar and not necessarily all the attributes. So in AutoPB, the user can choose what attributes that they would want to import from external sources.

5.1 Import module

![External sources integrated in AutoPB](image)

Figure 5.1: External sources integrated in AutoPB

The user can import from different external sources shown in Figure 5.1. This is an additional functionality the system provides apart from entering data manually. The attributes that can be imported from these sources are given below:

5.1.1 Google Scholar

Google Scholar is a freely accessible web search engine to search for publications or scholarly literature across multiple disciplines. Using Google Scholar user can search scholarly literature related to a particular topic as well as search for publications of a researcher. We use Google Scholar as one of our external sources to import:

- Publication
- Picture
While importing data from Google Scholar we faced the name ambiguity problem. Author name ambiguity problem is a well known problem that arises when linking an author to a scholarly article. When there are more than one author with the same name, its hard to match them with their respective publication. This is an entirely different problem that many system like ORCID aims to solve it. To resolve this problem in AutoPB, when a user wants to import an attribute from Google Scholar we asked the user their username so that we can easily link their Google Scholar profile. As Google Scholar does not have any APIs that we can use in order to access the data, we created our own crawler and parser. The crawler first crawls the entire profile of the user and parses it in order to get attributes and its fields.

![Google Scholar profile for Professor Jiawei Han](image)

**Figure 5.2:** Google Scholar profile for Professor Jiawei Han
5.1.2 NSF

National Science Foundation (NSF) maintains their own database of the list of awards provided by them. NSF provides a public API that can be used to fetch the award details awarded to a researcher. We use this API in AutoPB to get the NSF awards received by a user. The API requires the full name of the user as an input and outputs the list of awards with their details received by the user. Using NSF API we fetch the Award attribute as shown in Figure 5.5.
5.1.3 Justia

Justia Patents is a website that maintains millions of US Patents which can be freely accessed. In AutoPB we import Patents from Justia making Justia one of the external source. In Justia Patent’s website we can search patents of a particular user by simply typing their name. While importing from Justia we leverage the same search functionality in their website. We pass the full name of the user as input to the search box and get the list of patents
returned by it. We can see in Figure 5.6 the list of patents returned while searching for Professor Jiawei Han’s patent. Similarly in Figure 5.7, we can see the Patents imported from Justia in our Profile Builder by parsing the page shown in Figure 5.6.

Figure 5.6: Patents returned from search in Justia website

<table>
<thead>
<tr>
<th>Patent Number</th>
<th>Title</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>200302925801</td>
<td>Methods and system for mining frequent patterns</td>
<td>This invention provides methods apparatus and data structures useful for mining databases for frequent items. The invention uses a frequent pattern tree to represent the contents of a database in a manner which is conducive to data mining. The frequent pattern tree tends to be smaller than the original database. A frequent pattern tree can be viewed recursively. The frequent pattern tree and associated methods and apparatus of this invention is relatively fast, efficient and scalable and can be used to mine both long and short frequent patterns.</td>
</tr>
<tr>
<td>201602030830</td>
<td>SAMPLING OF USERS IN NETWORK A/I TESTING</td>
<td>The disclosed embodiments provide a system for performing network A/I testing. During operation, the system obtains a graph of a social network and calculates a set of equally sized clusters of users in the social network by iteratively switching memberships of the nodes among the equally sized clusters to increase a number of edges in each of the equally sized clusters. Next, the system randomly selects a subset of the equally sized clusters for exposure to a treatment version of a message. The system then performs an A/I test by presenting the treatment version to the selected clusters and tracking a response of the selected clusters to the treatment version.</td>
</tr>
<tr>
<td>20120904184</td>
<td>Systems and Methods for Detecting a Novel Data Class</td>
<td>Systems and methods for data classification and novel data class detection are provided. In one illustrative embodiment, a system or method for detecting a novel class includes receiving a data stream comprising a plurality of data points, and identifying a set of filtered outliers, in the plurality of data points, that are outside of a decision boundary. A cohesion and a separation for the set of filtered outliers may be determined. A novel class may be detected using the cohesion and the separation of the set of filtered outliers, and the novel class may include the set of filtered outliers.</td>
</tr>
<tr>
<td>6655389</td>
<td>Methods and system for mining frequent patterns</td>
<td>This invention provides methods apparatus and data structures useful for mining databases for frequent items. The invention uses a frequent pattern tree to represent the contents of a database in a manner which is conducive to data mining. The frequent pattern tree tends to be smaller than the original database. A frequent pattern tree can be viewed recursively. The frequent pattern tree and associated methods and apparatus of this invention is relatively fast, efficient and scalable and can be used to mine both long and short frequent patterns.</td>
</tr>
</tbody>
</table>

Figure 5.7: Patents from Justia imported in AutoPB
5.1.4 Science Direct

Science Direct is a website which provides access to scientific, technical, and medical research. According to Wikipedia it is known to hosts over 12 million pieces of content. In AutoPB we have integrated science direct as one of the source that the user can import their Publications from. Science Direct provides an elsevier API to search the Science Direct database for Publications of a user. We use this API to fetch the publications from Science Direct which takes full name and affiliation as input and returns a list of Publication in JSON format. Figure 5.8 shows the publication imported from Science Direct in our Profile Builder using the elsevier API.

![Figure 5.8: Publication from Science direct imported in AutoPB](image)

5.1.5 LinkedIn

LinkedIn is a famous social networking website used by researchers as well as working professionals to share their portfolio and make professional connections. We integrated LinkedIn in our Profile Builder to import the following attributes:

- Profile pic
- Current Employment
- Current School
- Address
- Employment
- Education
- Summary
Research Interest

Importing from LinkedIn was very challenging because unlike other website it didn’t provide any APIs or search functionality. One of the option we considered was to crawl the public profile of the user. But this method was restrictive as some users might not have their profile public or might not want to make their profile public. Other than that public profile only contains limited profile information of the user and not the full profile of the user. So, we decided to import the private profile of the user. As LinkedIn does not provide any API to fetch the private profile of the user even though the user is signed in, we decided to create a google chrome plugin which would crawl the profile of the user and send it our system. We will discuss more about the plug in the next chapter. So, in AutoPB when a user wants to import their profile from LinkedIn we ask the user to sign in to LinkedIn and go to their profile page and click the plugin icon so that the page is crawled and sent to our Profile Builder. We crawled the profile of the user and parsed it to get the above mentioned attributes of the user. Figure 5.9 shows the Experience and Education in the LinkedIn profile. Whereas, Figure 5.10 shows the Education and Experience imported in our Profile Builder from LinkedIn.
Figure 5.9: Education and Employment shown in the LinkedIn profile

Figure 5.10: Education and Employment imported from LinkedIn
5.1.6 Videolectures.net

VideoLectures.net is the world’s biggest academic online video repository. We integrated AutoPB with videolectures.net so that the users can import their lectures in their profile. Videolectures.net website contains a search functionality in which we can search using the full name of the person to get their lecture videos. We leveraged the same search functionality to get list of video lectures of a user. We parse the search results in order to get the respective attributes as shown in Figure 5.12 in our profile builder.

Figure 5.11: Video lectures returned by the search result in Videolectures.net website

Figure 5.12: Video lectures imported from Videolectures.net in AutoPB
5.2 Export module

After the users import and edit all of their records, AutoPB provides different options to export described below:

- **Online Profile**: A customized online profile is created that the user can share with others.

- **Resume (PDF)**: User can download the customized resume in PDF format.

- **API**: We also provide API so that users can call the API to get the edited records to embed it in their personal website. The API takes username and set of attributes as input. The output is in JSON format as shown in Figure 5.2.

![JSON Output Example](image.png)

**Figure 5.13**: API output for Jiawei Han showing Honors
CHAPTER 6

CLIENT-SIDE SEARCHING, CRAWLING
AND PARSING

Client-side searching, crawling and parsing is one of the main objective while building our Profile Builder. When we import from external sources all the crawling and parsing is done in the client-side. Performing the search and parsing in the client side has many advantages compared to doing it on the server-side. Some advantages are listed below due to which we decided to perform client side data retrieval:

- Some external sources provide APIs that users can use to fetch data. But most of the API will have some limit on how many request a user can send in a day/hour. If the API is called in the server side then the system might block a request whenever it exceeds the limit. By calling the API from the client side we won’t be facing this problem since each request will be from different IP address.

- As the number of users of the system increases the load in the server side increases as well. If the load increases in the server there might be delay to process the request. By performing the search, crawling and parsing in the client side we are reducing the load from the server. Therefore, the server can be used for other computational expensive task making it easier to scale up the system.

We implemented the client side search using Node js. There are 2 Node js packages mainly used while developing the crawler and parser in AutoPB:

- Request: Request is designed to be the simplest way to make http/https calls. It supports HTTPS and follows redirects by default which makes it easier to use.

- Cheerio: Cheerio is used to parse the page fetched by the request call. We send the output given by the request method which will be the
DOM of the page and pass it to cheerio. Cheerio reads the DOM structure of the page and makes it easier to parse the page to get the attributes and fields needed.

Node.js is a popular server-side scripting language. After writing the crawler and parser in node.js the next step was to convert the code into browser compatible form so that the browser can run the script. In order to do this we used a library called Browserify. Browserify is a tool used for compiling node flavored commonjs module for the browser. Using browserify we can include the 2 modules mentioned above: require and cheerio and build a bundle that can be served by the browser in a single script tag. Browserify supports most of the npm installed libraries to convert it into browser compatible form.

While requesting for a page or while calling an API from the browser we faced some problems. As some services didn’t allow cross origin resource sharing (CORS) the browser was unable to get the data. CORS defines a way in which browser and server can interact. CORS allows more freedom and functionality compared to same-origin request in which a request is only addressed if it comes from the same domain where the server resides. Some APIs that we were calling as well as some pages that we were crawling had same origin request policy. So, while fetching the data from those service the browser would get an empty result. In order to solve this problem, we created a browser extension that the user can add. The extension adds the header to the fetched data in order to allow CORS so that the browser can fetch the results.

External sources that have public APIs or public profile were easily crawled and parsed using the method above. But for LinkedIn the process was challenging since it required the user to log in. We tried couple of methods to integrate LinkedIn but failed to do so. First, we tried similar method mentioned by writing the node.js script to crawl the private profile. We used selenium-web driver for this which automates the process of login and redirects to the profile of the user. Then, we fetched the user profile and parsed it to get the attributes. But while converting this code through Browserify it failed. It failed because selenium web driver could not run in the client side i.e. from the browser. This limitation comes because of many security reasons. Because if browser automation is enabled it would be a security
concern in the client side as anyone can run any script from your browser without your consent. I tried using other equivalent Browserify libraries but till date none of them supported web drivers.

The next method we tried was using iframe. Using iframe I wanted to incorporate the LinkedIn website in our Profile Builder. The idea was after the user signs in to LinkedIn then we can redirect it to the users Profile and fetch their data from the iframe. But since the domain its hosted is different from linkedin domain. The sign in process could not go through. So this attempt was unsuccessful as well.

Another method that I used was simply using javascript and ajax to crawl the profile of the linkedin after they are signed in. But as each request was hosted as a different session each time the public profile was crawled instead of the private profile because each session was independent of each other. I tried to see if there was any way that we can fetch data from another tab in Google Chrome from the current tab. If its within the same domain google chrome supports it but if it is a different domain its not possible. This is also because of security reasons as for example is a user is signed in to their bank account in another tab, if I can easily access their information from our profile builder that would be a major security threat.

After much research, we solved the LinkedIn crawling problem by creating a Google Chrome extension. We will be describing about this extension in detail in the next section below.

6.1 Google Chrome extension for LinkedIn

We can find many google chrome extension in the application store which helps in crawling a particular site. Most of them crawl the page and save the result in an excel sheet or in json format. As our main objective was to make the process client side involving the server was not an option for us. While creating a google chrome extension the major hurdle was to pass the crawled data to our Profile Builder without involving the server.

We designed the extension in a way that when a user logs in to LinkedIn and open their profile, when they click on the extension icon we get the DOM of the current page. After getting the DOM of the page we use the cheerio library mentioned above to parse the page to get the attributes that we need.
After getting the attributes and all their data in JSON format, the next step was to pass it to the Profile Builder. One option that I considered was to pass it to the server and let the profile builder fetch from there. But this process involved the server which was not a good option. Another option was to store it in the local storage of the browser but the data stored on the local storage could only be accessed by the same domain web page. So, saving the data in the local storage of LinkedIn could not be accessed from the Profile Builder website.

Google chrome extension has a content script which can access the DOM elements of a page as well as access the local storage of other website. We used this content script to get the DOM elements of the page. After getting the DOM elements of the page and passing it to the parser, we create an inactive tab of our Profile Builder using our extension. Using this tab we execute a script that stored the parsed JSON object to the local storage of the Profile Builder website. The JSON data was saved in the local storage of the Profile Builder website and could be easily accessed by any of the pages of Profile Builder. This way, the data from LinkedIn page was passed to our web page without involving the server using the Google Chrome extension.
CHAPTER 7

SYSTEM DEMO

In this chapter, we will demonstrate screen shots of our system showcasing the overall flow of AutoPB.

Figure 7.1: Homepage of AutoPB where user inputs name and affiliation

Figure 7.2: AutoPB when a new user uses the system
Figure 7.3: User Interface to search for existing attributes along with their schemas

Figure 7.4: User Interface to add new attribute or existing attribute with new schema
Figure 7.5: User Interface to add attribute from LinkedIn

Figure 7.6: Shows the Google Plugin in the top right corner of the window. When the user clicks the icon the data is fetched from the current LinkedIn profile page
Figure 7.7: User Interface after adding and importing all the attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Jane Hsu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Position</td>
<td>UIUC</td>
</tr>
</tbody>
</table>

- Patients
- ScienceDirect-Publication
- Videos
- NSF-Award
- GoogleScholar-Research-Interest
- GoogleScholar-Current-Position
- GoogleScholar-Publication
- Skills
- GoogleScholar-Picture

Add attribute

![User Interface after adding and importing all the attributes](image)

Figure 7.8: User Interface to edit records, add or delete fields

<table>
<thead>
<tr>
<th>Title</th>
<th>PublicationName</th>
<th>Date</th>
<th>Abstract</th>
<th>authors</th>
<th>Edit</th>
<th>Del</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Framework for Clustering Evolving Data Streams</td>
<td>Proceedings 2003 VLDB Conference</td>
<td>2003</td>
<td>This chapter discusses a framework for clustering evolving data streams. The clustering problem is a difficult problem for the data stream domain. This is because the large volumes of data streaming...</td>
<td>Chan C Aggarwal, Philip S Yu, Jawaher, Jeryong Bang</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Framework for Projected Clustering of High Dimensional Data Streams</td>
<td>Proceedings 2004 VLDB Conference</td>
<td>2004</td>
<td>This chapter presents a new framework, HFS, for high-dimensional projected clustering of data streams. It finds projected clusters in particular subsets of the dimensions by maintaining condensed...</td>
<td>Chan C Aggarwal, Jawaher, Jeryong Bang, Philip S Yu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add ScienceDirect-Publication

- Videos
- NSF-Award
- GoogleScholar-Research-Interest
- GoogleScholar-Current-Position
- GoogleScholar-Publication
- Skills
- GoogleScholar-Picture

![User Interface to edit records, add or delete fields](image)
Figure 7.9: User Interface to select attributes for the user’s profile

Figure 7.10: User Interface to select data
Figure 7.11: User Interface showing paginated data sorted by PaperYear

<table>
<thead>
<tr>
<th>PaperTitle</th>
<th>PaperYear</th>
<th>PaperConference</th>
<th>PaperAuthor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Graph-Based Consensus Maximization Approach for Combining Multiple Supervised and Unsupervised Models.</td>
<td>2013</td>
<td>IEEE Trans. Knowl. Data Eng.</td>
<td>Jing Gao, Yang Liang, Wei Fan, Yihao Sun, Jiawei Han</td>
</tr>
<tr>
<td>A Text Cube Approach to Human, Social and Cultural Behavior in the Twitter Stream.</td>
<td>2013</td>
<td></td>
<td>Xiong Li, Katch Tam, Jeffrey Hancock, Jiawei Han, ...</td>
</tr>
<tr>
<td>Efficient Deep Mining of Selective Discriminative Patterns for Classification.</td>
<td>2013</td>
<td></td>
<td>Hong Cheng, Jiawei Han, Xiaoli Yan, Philip S. Yu</td>
</tr>
<tr>
<td>Exploring structure and content on the web: extraction and integration of the semi-structured web.</td>
<td>2013</td>
<td></td>
<td>Tim Weninger, Jiawei Han</td>
</tr>
<tr>
<td>Feature selection using dynamic weights for classification.</td>
<td>2013</td>
<td>Knowledge-Based Syst.</td>
<td>Xin Sun, Yingheng Liu, Xintao Xu, Kui-Ling Chen, Jiawei Han, ...</td>
</tr>
<tr>
<td>On Exploiting Transient Social Contact Patterns for Data Forwarding in Delay-Tolerant Networks.</td>
<td>2013</td>
<td>IEEE Trans. Mob. Comput.</td>
<td>Wei Gao, Quanhong Cao, Tom Lu, Peter, Jiawei Han</td>
</tr>
<tr>
<td>Reinforced Similarity Integration in Image-Rich Information Networks.</td>
<td>2013</td>
<td>IEEE Trans. Knowl. Data Eng.</td>
<td>Xun Jiu, Junjie Liu, Jie Yu, Gang Wang, Jiawei Han, ...</td>
</tr>
<tr>
<td>Trustworthiness analysis of sensor data in cyber-physical systems.</td>
<td>2013</td>
<td>J. Comput. Syst. Sci.</td>
<td>Lu An, Tong, Xiaoyu Su, Songying Kian, Quanquan Gu, Jiawei Han, ...</td>
</tr>
<tr>
<td>A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration.</td>
<td>2012</td>
<td>CuPPI</td>
<td>Bo Zhao, Benjamin H.立刻, Rubirola, Jim Gemmel, Jiawei Han</td>
</tr>
<tr>
<td>A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration.</td>
<td>2012</td>
<td>PKDD</td>
<td>Bo Zhao, Benjamin H.立刻, Rubirola, Jim Gemmel, Jiawei Han</td>
</tr>
</tbody>
</table>

Showing 1 to 10 of 661 entries

Figure 7.12: User Interface showing template selection where the user can select from existing templates
Figure 7.13: Output resume in PDF format
In this thesis, we introduced AutoPB that consists of 3 main concepts 1) Schema sharing, which aims to share schema of attributes among users 2) Flexible import and export, which provides users the flexibility of importing from different external sources and export profile in different forms 3) Client side data retrieval, which aims to reduce the load in the server side by performing searching, crawling and parsing in the client side. The proposed Profile Builder benefits include reduced reporting workload, improved attribution and better support for managing online profiles. We demonstrate that the system acts like a one stop portal for researchers to create customized online profile/resume. We also provide API’s that the user can use to integrate the system in their personal homepage.

In the future, we can include feature to import attributes from user’s current resume. Another feature that can be added is auto syncing with different external sources so that the user don’t have to manually update each time they sign in. Automatically merging attribute from different sources can also be explored. For example if Publication is imported from Science Direct and Google Scholar both, we can have a functionality that combines the data from both these sources.
REFERENCES


