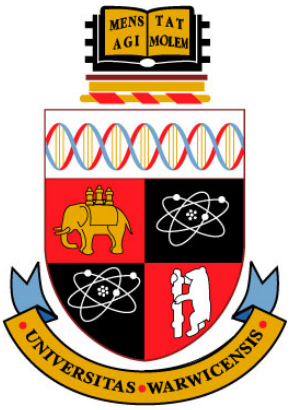
APPENDIX

ONE

SPECIFICATION REPORT



Online User Privacy Investigation Using Social Profile Seeding

CS310 Computer Science Project

Project Specification

Adam Coles

Supervisor: Dr. Matthew Leeke

Department of Computer Science
University of Warwick

2016-17

Contents

1 Introduction

With access to the internet becoming increasingly vital in the modern world it is of no surprise that globally, as of June 2016, 49.2% of the population are considered to be active internet users [12]. Across all users social media is the most popular activity to engage in whilst online, and a staggering 31% of the global population own a profile on at least one social platform [22]. All this activity produces a vast amount of personal, potentially open data, with Facebook alone currently holding over 300 petabytes of information across their warehouses [25]. With an immense amount of information available on the internet your privacy becomes difficult to maintain.

1.1 Motivation

The term 'active digital footprint' refers to the personal data an internet user gives permission to be accessible online, and all users of social media will have one. As the number and purposes of social media expands this digital footprint becomes more detailed, allowing users to connect with new people who share similar interests or friendship circles. However, people are often unaware of just how much data they make available for the world to see, a fact utilised by the police to track down criminals and known associates [1] [24]. A larger social media presence gives more chance to find additional information about a person that they have not explicitly shared. This includes public records, which have transitioned from paper records to a digital format, and that have been criticised for giving away overly personal details [3]. Whilst the police use open data to benefit society, if they have that ability so do potential criminals or stalkers, and social media consumers must be aware just how much can be found out about them online [18].

Despite attempts to make people aware of the dangers of open online profiles, in general there is a blasé attitude towards online privacy. In 2012 a survey showed that 26% of American Facebook users shared their entire profile publicly, including all wall posts [17]. Without being scared into a realisation of how important privacy is, there will not be a change in attitude towards this issue.

1.2 Project Aims

The primary goal of the project is to create an online profiler that uses a Facebook profile as a seed in order to collect as many details as possible about a user from online open data sources. Whilst their will be some complex algorithmic design in order to speed up the process, the majority of aims focus on effective internet crawling. Results from the project should assist in a variety of areas, from law enforcement to social media awareness.

1.2.1 Locating Social Media

Even though some people will not have a wide selection of public profiles, using Facebook as a starting point assists in locating as many accounts as possible owned by an individual. All this data can be deemed to be personal and can be aggregated to create concrete assumptions by correlating truths between them. Depending on the number of accounts, and their privacy settings, limits the amount of data available for collection. For some extreme examples, personal truths can be drawn from the information given, for instance geo-tagged tweets on Twitter. Some websites

provide detailed API that should provide all the data required, although others will require the webpage to be parsed in more complex ways. Research must be done to ensure the most efficient and effective data harvesting method is used.

1.2.2 Connecting to New Sources

Once some social media has been examined more internet sources need to be found, such as the previously mentioned public records. A simple Google search with a handful of parameters can find obscure references about a person that may either solidify assumptions or create new theories that can be explored. This area of search is more likely to show unexpected results, as the target is in less control of the privacy of data displayed. The main drive of this aim is that the searches are recursive, that is if new information is found and confirmed the search should reoccur with additional parameters.

1.2.3 Receiving Real-Time Input

In some cases a human can confirm faster and with greater certainty specific details about data found. If, for instance, the profile picture of an individual is of them wearing a police uniform, and the system suspects they are police officer, a human can confirm the assumption, allowing the system to use this suspicion as fact. Not only does this speed up the process significantly, adding this feature reduces the complexity of algorithms used whilst improving the accuracy of the results. Decisions will have to made to restrict the amount of user input requested, as the system is meant to be for the most part automated.

1.2.4 Using Open Data and Open Software

One core value that must be upheld during the project is that all data used is open and any third-party software used is free to license. This not only ensures the privacy of test subjects it also ensures that there is no budget to development and production. End users of the system will typically not have the funds to pay for complex software or hardware, therefore it is essential that as little money is spent as possible.

1.3 Stakeholders

The primary stakeholders in the project are the developer and the project supervisor. Any people who give their permission for their data to be used whilst testing the system will have to have their privacy guaranteed, in case secret information is found. The opinion of a law enforcement officer on the final outcome is also desired, as they would be the primary user of a system similar to the one being developed.

2 Related Work

As mentioned, the use of someones online footprint to learn about them and their associates is already occurring within law enforcement [24]. This topic has also been a feature of many academic

studies, particular when looking at teaching others to manage their online image [6]. Existing systems will have a heavy influence throughout the project, as these are direct competition to the final developed product, and provide a good testing base line. They also assist when choosing third-party software and creating the complex algorithms that are necessary to find hidden information. Meanwhile scholarly work into this area will help when it comes to understanding the effect the results of this project will have on social and legal issues.

2.1 Collecting Information

It is likely that anybody who is interested in finding out anything about a person using the internet will turn to a search engine. In 2013 Google had approximately 30 trillion websites indexed, so even searching using detailed parameters produces thousands of results [16]. However, search engines are not the only way to scour the web. Customisable or specialised web crawlers such as Apache Nutch, a Java based modular crawler, can output more desirable results than Google.

Each social network has a different level of access when it comes to harvesting user data. Russell (2014) explored many techniques for mining certain sources, including friendship graphs on Facebook and Twitter frequency analysis [20]. Even when inside social media, crawling is still necessary to explore all possible links in a graph, as few transitions from page to page can be hundred percent guaranteed to be useful. For instance, just because two people are friends on Facebook does not necessarily mean they know each other, particularly in less educated areas. O'Dea and Campbell (2012) found that 13.5% of Australian adolescents accepted unknown friend requests [5].

For specific open source web crawlers, *Apache Nutch* is highly scalable, robust Java web crawler. As it is built upon the *Apache Lucene* framework, Nutch is modular and therefore can be combined with other packages to expand upon the features available, such as search or indexing with *ElasticSearch* [10]. *Norconex* is another Java based crawler with similar functionality to Nutch. It comes with a plethora of documentation and a handful of tutorials, including how to crawl with the assistance of Facebook API [8]. However, due to being less established than Nutch, *Norconex* has less support from other tools and is not maintained as well.

A web crawler locates websites and retrieves the HTML, yet to receive any meaningful results this data must be indexed and searched. *ElasticSearch* is the most popular search engine, currently ranking above its similarly Lucene based counterpart *Apache Solr* [7]. Both can be run within Java with HTTP and JSON interfaces, and for the most part are very similar, with notably *ElasticSearch* leaning more towards REST APIs. However *ElasticSearch* has a more robust aggregation feature, and therefore may be more suitable for the project [23].

Building on this further, when fed a large block of text the system may need to understand natural language to find key data. Many natural language processors exist for this function, but few are open source. *Semantria* for instance is primarily a sentiment analyser. A set of standard API is provided and a subscription is paid to have access to the *Semantria Cloud*, giving the benefit of high-speed, scalable performance. IBM's Watson also has in built sentiment analysis along with a range of other text manipulation modules. Whilst immensely powerful, some features of Watson have been replicated before to relative success within a university classroom [26], giving hope that as an extension these features can be simplified and added to the scope of this project.

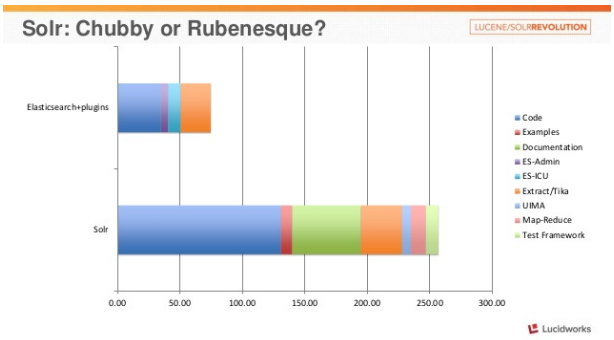


Figure 1: Comparing the Size of *ElasticSearch* Against *Apache Solr* [19]

2.2 Creating the Profile

Once data has been collected some means of aggregation is required to produce meaningful results. That is the data has to be compared to find correlations, the source of the data has to be examined for trust, and assumptions have to be made.

Within the majority of social networks, especially media platforms, there is a recommendation engine. While their outputs are not that desirable for the scope of this project, their means of gathering and processing information on a user could be helpful. As elegantly put by Finger (2014), a recommendation engine "reduces Big Data to small data", a definition that could easily fit with the aims of this project [9]. Netflix value their recommendation engine highly, with claims 75% of users watch from recommendations [2].

Table I: Recommendation Techniques

Technique	Background	Input	Process
Collaborative	Ratings from **U** of items in **I**.	Ratings from **u** of items in **I**.	Identify users in **U** similar to **u**, and extrapolate from their ratings of **i**.
Content-based	Features of items in **I**	**u**'s ratings of items in **I**	Generate a classifier that fits **u**'s rating behavior and use it on **i**.
Demographic	Demographic information about **U** and their ratings of items in **I**.	Demographic information about **u**.	Identify users that are demographically similar to **u**, and extrapolate from their ratings of **i**.
Utility-based	Features of items in **I**.	A utility function over items in **I** that describes **u**'s preferences.	Apply the function to the items and determine **i**'s rank.
Knowledge-based	Features of items in **I**. Knowledge of how these items meet a user's needs.	A description of **u**'s needs or interests.	Infer a match between **i** and **u**'s need.

Figure 2: Recommendation Engine Techniques [4]

Unlike websites with a recommendation engine, the proposed system has no access to private

information. Amazon, for example, store data on all products a user views, and from there can make educated decisions on other products of interest. As noted by Iskold (2007), Amazon relies on remembering what you've done "years and minutes ago", information not available for public viewing [15]. Despite this the aggregation techniques used are invaluable. Figure 2 shows Burke's (2002) descriptions of these techniques, which will provide a baseline for the assumptions made by the system [4].

There have been attempts at creating a social profile from crawled data in the past, unrelated to recommendation engines. Van Hinsbergh (2015) created a search engine which would attempt to create a profile of a person from a set of provided keywords and a useful starting seed [13]. Similarly to the proposed system, Van Hinsbergh's final application also took user feedback into account and used open data. Generously, Van Hinsbergh has given access to his report and some source code that will be referenced throughout development.

The proposed system is slightly different to other programs already available primarily due to the social media seed that will be the root of a person's profiling. This reduces the amount of manual searching required prior to utilising the application. User feedback will also be integral to the system, distinguishing it further from current applications.

3 Ethical, Social, Legal and Professional Issues

In every software development project certain standards must be maintained to ensure not only the end system is allowed to launch but also to solidify trust between developers and the end users. Stakeholders also require reassurance that their potential investment will not go to waste, or that they protected from any backlash due to improper procedures. The aim of this section is to discuss problems that may arise the ethical, social, legal and professional fields, and how preparations have been made to resolve them. For a base during development the British Computing Society Code of Conduct will be maintained, to protect customers and developer integrity [21].

3.1 Ethical Issues

Due to the highly personal nature of the intended system, a plethora of ethical issues emerge, the key issue being the privacy of the users testing the system. In order to keep the identities of the users safe, no personal data should be stored permanently after searching. This also eliminates the need for complex encryption when handling the information. Furthermore, any test users will have to give express permission for their profiles to be examined, and if they are selected to be an example for displaying the system's potential, they must give further permission for their details to be shown to stakeholders or other third-parties. At any point the user has the right to remove themselves as a test case if they feel uncomfortable with the information found.

3.2 Social Issues

Continuing from the protection of privacy of individuals using the system, there is the chance that some evidence found may lead to somebody feeling victimised. To prevent this as much as possible, the program will try to avoid presuming a persons sexual, religious or political preferences. Even though some of this data may not be private for all users, its best to abstain across the entire consumer range to avoid accidental discovery. On top of this, the system will avoid stereotypes regarding race or disability entirely, and there will be no bias assumptions made from these. Should the system be successful in its aims and locate personal data from inputted profiles the view on social media may change for some users, hopefully leading them to be more aware of profile security.

3.3 Legal Issues

Since the final program will inevitably use some third-party software or libraries the developers must ensure that they have the correct licensing. A major aim of the project it to only use open data and open software so all third-party sources should be covered by some open source license [14]. As personal data will be mined and stored at some time during the program operation, the system must comply to the terms of the Data Protection Act [11]. Following from this, the data collected must come from legitimate sources that have the right to own the data to begin with, although this may be hard to verify.

3.4 Professional Issues

It is critical for the platforms success that a user finds the data presented to be valuable and that they trust the system. As mentioned, data will only be temporarily stored to try to guarantee information found is only seen by the intended user. Also aforementioned developers will follow the BCS Code of Conduct which ensures they are working for the public's interest and for the profession [21].

4 System Requirements

Breaking down the main aims of the project into detailed requirements allows progress of software to be tracked throughout the course of development. These requirements may exactly represent the final system as they have room to change over time however they provide a strong base to work on.

4.1 Functional Requirements

The following requirements look at what the system can do; the inputs it receives, how these inputs are processed and the outputs it produces:

F1: *The system must allow users to search for information using a Facebook seed (profile).*

F2: *All available data should be mined from the Facebook input and then from other social media should a link be established.*

F3: *The system must recursively search for more data once a detail on the individual has been discovered, across the entire internet.*

F4: *The data mined by the system should be related only to the inputted profile in some way however tenuous*

F5: *The system should ask the user for feedback if an assumption has been made without full certainty and human decision would benefit.*

F6: *The system must avoid overly sensitive data when searching for information.*

F7: *The system should avoid elaborate inferences that both it and a human will have difficultly verifying.*

F8: *The system should have a clear interface to display the data, including a simple search bar for the Facebook link and some area to ask for user input during the search.*

F9: *The system should allow the user to define the depth of the search, allowing for a quick search to find the basics and more robust search to find as much as possible.*

F10: *The user should be apply to supply additional information to the system if it is already a known fact, giving the system a further head start when searching.*

4.2 Non-Functional Requirements

The following requirements look at the attributes of the system; how it will perform in a real-world scenario:

NF1: *The system should be extensible. Features should be implemented as modules, that can be easily added to data pipelines.*

NF2: *The system should be scalable. If moved onto a faster machine, the system should be able to locate and process information at a faster rate.*

NF3: *The system should be efficient. Even though search may be time consuming the user should be updated with information as soon as it is available.*

NF4: *The interface to the system should be easy to understand and provide information in a clear and unambiguous format.*

NF5: *The system should be maintainable, through both well documented and well presented code, so future development can be made if required.*

4.3 Hardware and Software Constraints

In order to keep the system accessible to its target users and to match the aims of the project some constraints must be made on the resources used. Then end program should be able to run on home computers as this typically the level used in law enforcement. There are no exact time restraints put on search response time, however it should be reasonable since the system is not fully automated, even on lower end machines. As mentioned previously, the software should use only open license sources, to add no extra costs to development and ownership.

4.4 Foreseeable Challenges

The internet is astronomically huge and is impossible to search fully. Deciding on the cut off point of search will be one of the main challenges of the project, hence some user input is allowed to give the system some idea of the level of detail sought after. The developer working on the project has no prior experience in web crawling, so there will be a learning curve to overcome before progress can be made on development.

5 Testing and Success Measurement

For any software project testing is critical in ensuring your end system meets the aims and requirements set out pre-development.

5.1 Testing Strategy

During development unit tests for each function and class will be produced to both make sure the outputs match what is expected and to catch the rare edge cases that may occur in a live run. Following this multiple related units will be combined together. A broader over-arching functionality of different areas of the code will be tested, for instance search or analysis. This is known as integration testing, attempting to find flaws in the information pipeline between units. All the modules will then be combined to create the final system tests, looking at the system as a whole. Ultimately theses tests will be directly related to the requirements set out in Section 3. Finally, and arguably most critical for this particular project, there will be user tests created to gauge the satisfaction levels of end users. Their feedback can be used to iteratively improve the system until everyone is happy with the results produced.

5.2 Success Criteria

Whilst in some projects simply hitting all the requirement is sufficient enough to claim it was a success, in many cases more details are needed. With this system user satisfaction is of the highest priority, and the user tests will be the main judge of the overall success, even though all requirements will still need to be met.

6 Project Management

To keep a software development project on track a plan is required to manage time effectively. This plan will be highly flexible to deal with sudden deadlines or potential unforeseen circumstances however will be a rough guide to follow during all stages of development.

6.1 Software Development Methodology

The software development methodology of choice is an agile approach, in documentation and the creation of the software itself. This easily allows for adaptations during development, should the aims or requirements of the project change. Also since there is a single developer allocating work is unnecessary so concrete decisions restrict rather than assist.

6.2 Design Approach

A top down approach to design will be used, that is each area of the system will be broken up into different modules, these modules designed and developed for the most part separately, and finally combined to create a larger system. This method makes the complete problem more manageable, although does mean it will take longer to have a working prototype. Despite this each module can still be tested using unit tests, a technique described in Section 5.1.

Before beginning work on any new section of the system a rough plan will be sketched, to avoid going out of scope or forgetting certain intricacies. To keep track of changes made, and to give room for features to developed in many ways, Git will be used as a version control software, with every commit being highly detailed. Comments will be used to explain complex parts of the code so a recall to memory is not required when it comes to writing the report.

6.3 Project Timeline

As the team developing this project consists of one person many agile methods are overly complex and would not be suitable, such as SCRUM. Instead, a simple adaptable Gantt chart (Figure 3) will be used to loosely track deadlines and give vague estimated times of the completion of certain modules in the system, as well as key project deadlines.

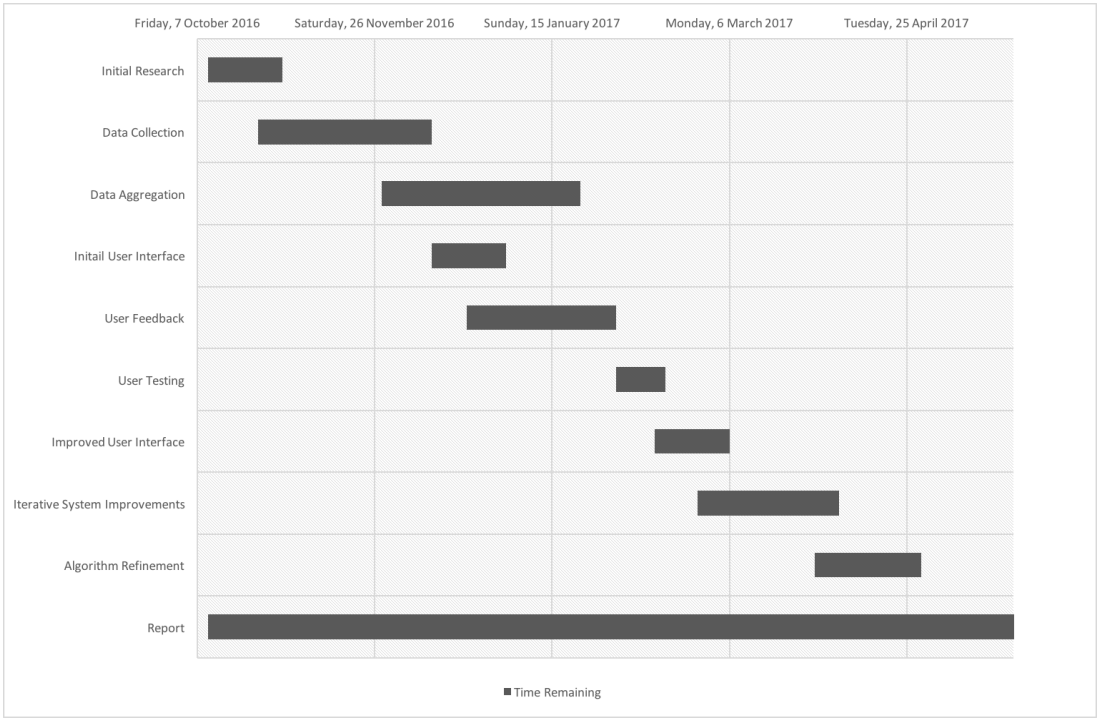


Figure 3: Gantt Chart Displaying Proposed Deadlines

6.4 Tools

During the project the following software tools will be used to aid in documentation, communication and development (Figure 4). Skype will be used so stakeholders can contact each other urgently when email will not suffice and a meeting cannot be made. LaTeX will be used to write any official

Figure 4: Selection of Development Tools' Logos

reports, as it produces a professional and easy to read document; TeXShop for Mac will hence be the word editor of choice. For code editing, a combination of Sublime Text and Vim will be used as this is the environment the developer feels most comfortable in. When sharing documents between multiple people is required DropBox will be used, since it provides a simple interface with notifications when changes are made.

7 Conclusion

To summarise the document, the core aim of the project is to create an online profile builder using social media as a seed. The main functions of the system have been outlined in Section 3.1, primarily the ability to locate different social media sources, mine them for information and to expand outwards to further sources on the internet. Over the next few months an application will go through design, development and testing, in hope of producing results that will change society's opinion on social media privacy.

References

[1] Susan B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2006.

[2] Xavier Amatriain & Justin Basilico. Netflix recommendations: Beyond the 5 stars (part 1). *Available at:* `http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html`.

[3] Kristen M. Blankley. Are public records too public? why personally identifying information should be removed from both online and print versions of court documents. *Ohio State Law Journal*, 65(2):413–450, 2004.

[4] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[5] Bridianne O'Dea & Andrew Campbell. Online social networking and the experience of cyber-bullying. *Annual Review of Cybertherapy and Telemedicine*, pages 212–217, 2012.

[6] Nicole Osborne & Louise Connelly. Managing your digital footprint: possible implications for teaching and learning. *European Conference on Social Media, Porto, Portugal*, 2015.

[7] db engines.com. Db-engines ranking of search engines. *Available at:* `http://db-engines.com/en/ranking/search+engine`.

[8] Pascal Essiembre. How to crawl facebook. *Available at:* `https://www.norconex.com/how-to-crawl-facebook/`.

[9] Lutz Finger. Recommendation engines: The reason why we love big data. *Available at:* `http://www.forbes.com/sites/lutzfinger/2014/09/02/recommendation-engines-the-reason-why-we-love-big-data/#7703e004218e`.

[10] Apache Software Foundation. What is apache nutch? *Available at:* `https://wiki.apache.org/nutch/FrontPage\#WhatisApacheNutch.3F`.

[11] GOV.UK. Data protection. *Available at:* `https://www.gov.uk/data-protection/the-data-protection-act`.

[12] Miniwatts Marketing Group. World internet users and 2016 population stats. *Available at:* `http://www.internetworldstats.com/stats.htm`.

[13] James Van Hinsbergh. Generating investigative profiles through open source data fusion. Technical report, University of Warwick, 2015.

[14] Open Source Initiative. Licenses & standards. *Available at:* `https://opensource.org/licenses`.

[15] Alex Iskold. The art, science and business of recommendation engines. *Available from:* **ReadWriteWeb**.

[16] Tune J. Koetsier. How google searches 30 trillion web pages, 100 billion times a month. *Available at:* `http://venturebeat.com/2013/03/01/how-google-searches-30-trillion-web-pages-100-billion-times-a-month/`.

[17] Consumer Reports Magazine. Facebook & your privacy: Who sees the data you share on the biggest social network? *Available at:* `http://www.consumerreports.org/cro/magazine/2012/06/facebook-your-privacy/index.htm`.

[18] Alice E. Marwick. The public domain: Social surveillance in everyday life. *Surveillance & Society*, 9(4):378–393, 2012.

[19] Alexandre Rafalovitch. Solr vs. elasticsearch - case by case. Lucene/Solr Revolution Conference, 2004.

[20] Matthew A. Russell. *Mining the Social Web*. O'Reilly Media, 2014.

[21] British Computing Society. Bcs code of conduct. *Available at:* `http://www.bcs.org/category/6030`.

[22] Statista. Global social network penetration rate as of january 2016. *Available at:* `https://www.statista.com/statistics/269615/social-network-penetration-by-region/`.

[23] Kelvin Tan. Apache solr vs elasticsearch. *Available at:* `http://solr-vs-elasticsearch.com/`.

[24] Daniel Trottier. Policing social media. *Canadian Review of Sociology*, 49(4):411–425, 2012.

[25] Pamela Vagata & Kevin Wilfrong. Scaling the facebook data warehouse to 300 pb. *Available at:* `https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/`.

[26] Walid Shalaby & Adarsh Avadhani Wlodek W. Zadrozny, Sean Gallagher. Simulating ibm watson in the classroom. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 378–393, 2015.