THE UNIVERSITY OF CHICAGO

ALUMNI INFORMATION COLLECTION AND MANAGEMENT

BY

Lingchen Lynette Dang

Apr 1 2023

A Paper Submitted in Partial Fulfillment of The Requirements for The Master of Arts Degree in The Master Of Arts in Computational Social Science

Faculty Advisor: Zhao Wang

Preceptor: Shilin Jia

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, Professor Zhao Wang, I am extremely grateful for all your guidance, advice, support, and critiques on this thesis. Your insight and expertise have been invaluable in molding this thesis into its final form. Thank you for being the most motherly, responsible, understanding, and nurturing thesis advisor one could ever ask for. Beyond this thesis, your encouragement and feedback have been instrumental in helping me achieve my academic and personal goals, and I am truly grateful for your mentorship and our friendship.

To Professor Sabrina Nardin, Professor Jon Clindaniel, and my preceptor Shilin Jia, although you didn't give direct feedback on my thesis, the knowledge and skills I gained from your classes have served as a stepping stone for my thesis. All of you have helped me develop a deeper understanding of research method, web scraping, data cleaning, data analysis, and network analysis. With your teaching, I was able to explore my interests and locate my passion. Besides, you are a constant source of inspiration throughout my studies.

To my friend Pranathi Iyer, thank you for collaborating on the industry classification algorithm with me during Computational Linguistics' final project. I want to acknowledge your contribution to the final paper, which serves as a precursor for an important part of this thesis. Your perspectives and insights helped me to broaden my understanding of pre-trained NLP models, and I am so proud of what we have accomplished together.

To Professor André Marques, my UI/UX classmates Wynne Tang, Abby Barnes, and MACSS administrator Shelly Robinson, thank you so much for your feedback on SSD Connect prototype. Your constructive criticism and thoughtful suggestions helped me to improve the design and functionality iteratively. I am truly fortunate to have such a talented and supportive community behind my back while I am writing this thesis.

Last but not least, I would like to extend a heartfelt thank you to my parents, my boyfriend Silvan Baier, my housemates Avery McLain and Elliot Delahaye, as well as my friends Enrique Arcilla and Theresa Lane. You positivity helped me to stay motivated during the most challenging times. Thank you for giving me emotional support when I need it the most and for being there for me throughout this journey.

ABSTRACT

To address the inefficiencies of the existing manual alumni information collection strategy, this thesis proposes an innovative pipeline to automate alumni information collection and identify the prominent alumni in the three MA programs under The University of Chicago's Social Science Division (SSD). This thesis utilizes computational methods such as web scraping, text cleaning, and natural language processing to collect and verify alumni data through Google API and Selenium. The preliminary analysis unveils critical revelation of the career trajectories and employment outcomes of SSD alumni. Additionally, this thesis proposes a machine-learning-driven approach for industry classification in mentor matching. As an extension of alumni data collection and management that establishes the groundwork for future integration of SSD's career service and alumni outreach platforms, a prototype of an all-in-one alumni management and networking web application, SSD Connect, is created. This thesis contributes to the body of literature by providing novel methods and insights for automating alumni information collection, managing and analyzing alumni networks, as well as optimizing alumni management and networking strategies for higher education institutions.

CHAPTER 1

INTRODUCTION

Alumni networks provide long-term value to an educational institution by giving alumni the chance to stay in contact with the school, the student body, and with each other (Ebert et al., 2015). First and foremost, alumni networks create and promote events, fundraisers, and donations to support ongoing projects, curriculum expansion, and scholarships for better student experience and job placements. Secondly, alumni network can help bridge the gap between former and current students, offering the student body academic mentoring and career coaching. Third, alumni can continue to learn from each other long after they have left school, to network with other alumni from different years, and to maintain an active, mutually beneficial relationships for lifetime.

With the technological advancement from recent years posing a threat to traditional manual alumni network management, a digitized alumni network is essential to support all type of interactions within and across stakeholders: the program administrators, the alumni, and current students. In this MA thesis, my goal is to identify and collect the information of the prominent alumni in the three MA programs under the Social Science Division (SSD): CIR, MAPSS, and MACSS. The existing data entries date back to 2000, including program, entry year, graduation year, citizenship status and undergraduate institution for 5,131 alumni in total. However, the current database has an incomplete coverage of alumni information and an untimely nature. Most information was collected prior or during the alumni's time at the University of Chicago, and failed to keep track of major updates in post-graduation years or the preferred way of contact for each alumni.

Instead of having an alumni outreach coordinator manually reach out to thousands of alumni, this thesis automates the process using computational methods such as web scraping, text

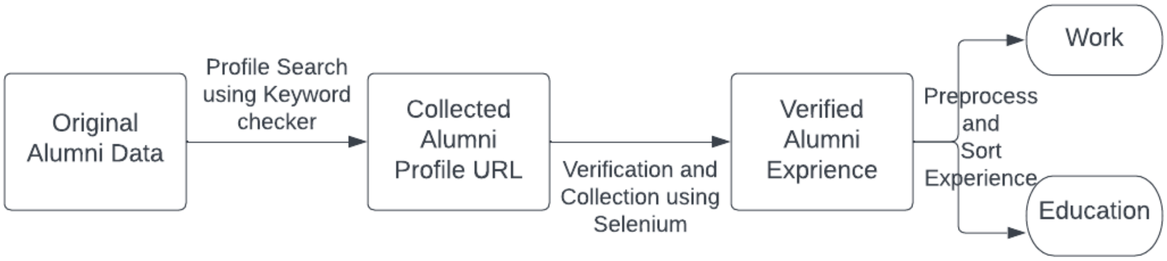

Original Alumni Data
Profile Search using Keyword checker
Collected Alumni Profile URL
Verification and Collection using Selenium
Verified Alumni Exprience
Preprocess and Sort Experience
Work
Education

Figure 1.1: Alumni Information Collection Workflow

cleaning, and natural language processing. The data collection process begins with web searching using Google API to automatically search and identify the profile of a matched person, followed by verification using keyword matching to compare the name, school, degree and program between the ground truth data entry we have and the information collected via Selenium. For each verified alumnus, I have collected key information, such as the current role, company, biography, and contact information using Selenium and BeautifulSoup. The collected information is stored, analyzed, and built into an alumni database that spans across the MA programs. Figure 1.1 shows the workflow regarding alumni data collection, verification, and preprocessing. With the collected data, the subsequent preliminary analysis shows that: (1) among the 618 verified alumni, 126 chose to pursue a PhD degree, 43 chose to pursue an MBA degree, and others went into the industry; (2) Education/Research is the most popular industry type, while researchers and managers are the most popular position types; (3) the job location of SSD alumni extends across the entire United States and multiple continents, with the greater Chicago area being the most popular location.

This thesis also attempts to classify alumni's work experience into different industry categories with machine-learning-driven approaches. I experiment with two different classification models: a pre-trained DistilBERT model and a logistic regression classifier trained on an external dataset. The findings indicate that the Logistic Regression classifier marginally outperforms the pre-trained BERT model, but both models need to be improved on the task of industry category classification. This can be attributed to various causes, including small and sparse dataset, inaccurate true labels, data shift in training and testing dataset, and homogeneity of feature variables.

Last but not least, I have created a prototype of the alumni networking web application as an extension of this MA thesis. The design is exclusively serving the alumni, administrators, and students of the three MA programs under SSD at The University of Chicago, to get connected, network, and seize the full extent of the programs.

CHAPTER 2
LITERATURE REVIEW

In this chapter, I will briefly review the evolution of formal and informal alumni networks, the existing professional alumni management platforms, and the core design principles for alumni management platforms.

2.1 Evolution of Formal and Informal Alumni Networks

Formal university-directed alumni networks have been in existence for decades and are constantly evolving throughout time. Traditional alumni networks are small in scale and highly reliant on university resources: they usually start from the existent friendships from pre-graduation years and the formation of regional alumni groups from post-graduation university fundraising activities and public relation events (Chi et al., 2012). These small networks continued to expand and gradually gained more importance in the development of universities because of their increasing outreach potentials and their added benefit to the school, the current student body, and the alumni themselves. On the one hand, integrating financial resources and human capital into alumni networks create extensive opportunities for donations, curriculum development, as well as new programs (Patras, 2020; Hall, 2011; David and Coenen, 2014). It is found that alumni networks in private universities make up on average 8.7% of the total budget of the universities in the United States (Altbach and Knight, 2007). As a result, alumni networks play an important role in the university business development as they support the introduction of new courses, the adoption of new curriculum, and the innovations of new programs.

On the other hand, alumni networks enable connections to form and strengthen between alumni and current students, and among alumni themselves. These connections can kindle a stronger sense of community, shape a collective alumni identity, and make alumni networks more robust. Alumni networks can help bridge the gap between alumni and current students, providing valuable opportunities for the current students to learn from their predecessor in various forms, including academic mentoring and career coaching (Wampler, 2013). These one-on-one interactions will set up a behavioral model of effective and long-lasting interpersonal communication and facilitate the integration of current students into the alumni networks when they graduate. On top of that, the alumni networks also bring alumni from different years and cohorts together, regardless of whether they have interacted in school (Gallo, 2018). The opportunities to interact with alumni from different years and cohorts open the door to new connections, more engagements, and strengthen relationships among alumni (Bardon et al., 2015). As a result, the alumni networks will create a sense of belonging and community and bring in a collective alumni identity (Aliberti et al., 2022).

Successful university-directed alumni networks have two core advantages: information and mobilization (Aarva and Alijärvi, 2012). However, in recent years, the advancement of technology forces the alumni system to undergo huge changes (Chiavacci, 2005; Barnard and Rensleigh, 2008). One of the contributing factors is the growing enrollment and the creation of new programs. They compound to an existing, large alumni base, resulting in difficulties or failures for manual management of vast information. Another underlying reason behind this transition is the decline of the traditional university-directed, manually-operated alumni networks as universities have lost their monopoly over the ownership of alumni information and the power to organize large-scale alumni events. Through social media, alumni can now organize events and activities on their own and form networks without any interference from universities.

Nevertheless, these informal alumni networks are very event-based instead of having a comprehensive, systematic structure, and have the potential to exclude certain groups of alumni based on preexisting social connections, geographical restrictions, and cultural differences (Shih, 2006). Consequently, universities and programs must re-strategize for formal university-directed alumni networks, make them more convenient and efficient, and ultimately increase their popularity to outperform alumni-directed informal networks. It is clear that automated and optimized features should be added to university-directed alumni networks. Therefore, the transition to an advanced, centralized, digitized alumni database and management system is essential to universities and programs (Mukherjee et al., 2019; Teixeira and Maccari, 2014).

2.2 Professional Alumni Management Platforms

In response to the need of automating and improving alumni database and management systems, many higher education institutions have adopted professional alumni management platforms to maintain relationships with their alumni, fostering a sense of community and encouraging ongoing engagement with the school (Barnard, 2007). Most of these platforms use cloud-based architecture with an intention to provide greater flexibility and scalability. However, the cloud-based nature also exposes these platforms to continuous, significant challenges in security (Brattstrom and Morreale, 2017). A huge advantage of cloud-based platforms is that they are light-weighted and versatile. They allow administrators to easily expand, update, modify, and customize their alumni management capabilities to meet their varying needs throughout time. But the downside of building cloud-based architecture is the rising prominent threats to data privacy. Currently, professional alumni management platforms use data encryption, authentication and access control, network security, regular auditing, and frequent updates to protect personally-sensitive alumni information against malicious hackers and prevent data breaches (Padhy et al., 2011; Almorsy et al., 2016).

In addition to cloud-based architecture, there are other innovative, distinctive features in professional alumni management systems. Most of them offer a convenient and accessible mobile version of the application in addition to a traditional web application (Barman, 2019). Without having to rely on a desktop computer or laptop, alumni are now able to stay connected with their alma mater and alumni network on the go from their mobile devices. With the help of a mobile app, it becomes effortless to keep track of updates through push notifications, RSVP to events, and manage conversation.

Professional alumni management systems typically include features such as contact management, event tracker, fundraising tools, and data analytics. Contact management features equip administrators with the latest update of alumni contact information, including phone numbers, email addresses, and physical addresses. Event trackers allow administrators to promote events and track attendance. Fundraising tools facilitate donations from alumni to support various initiatives at the universities, such as scholarship funds, new academic and extracurricular programs, as well as building and renovation projects. In the form of dashboards and reports, data analytics can be helpful for institutions to track alumni engagement, giving, and other key metrics. Other trendy features commonly found in alumni management platforms are job boards, alumni directories, and networking functions. In addition, some systems also offer integration with popular social media platforms like Facebook, Twitter, and LinkedIn, allowing alumni to connect and stay engaged with their alma mater on external platforms that complements the original platform.

Two of the most widely-used and established alumni management platforms are iModules and Graduway (Sağbaş et al., 2018; Stephenson and Yerger, 2015). Since 2002, iModules has been used by over 1,200 educational institutions, including universities, colleges, and K-12 schools, as well as nonprofit organizations. Based in the United States, it is known to be the preferred alumni management software of many Ivy League schools, including Yale University, Cornell University, Dartmouth College, and Harvard University, as well as some prominent, top-ranking US colleges such as Duke University and Stanford University. Founded in 2009, Graduway is a platform that supports a wide variety of colleges, universities, professional schools, K-12 schools, as well as nonprofit organizations in the United States and all around the world. While iModule's software focuses on alumni directory, email marketing, analytics, and fundraising, Graduway offers a wider range of features, on top of all the existing features from iModule, and has an emphasis on career development and mentoring, which are new to alumni management platforms and largely fosters alumni engagement.

In the past years, both iModule and Graduway have helped universities improve alumni engagement and foster strong relationships. However, in recent years, there has been a shift in the market towards modern, comprehensive alumni management platforms like Graduway. One reason for this is institution's desire for a unified structure for alumni management platform that can integrate easily with other school systems. Having all alumni resources in one place like Graduway makes it easier for them to manage and engage with their alumni community. Furthermore, the need for data analytics and insights has become possible with Graduway. With real-time, more advanced tracking and reporting features, institution can easily examine alumni behavior and preferences, which can inform their alumni engagement strategies and help target their outreach efforts more effectively. Last but not least, Graduway and other modern alumni management softwares offer more customized options and flexibility than older alumni management systems like iModule in terms of branding, features, communications, and notifications. This allows universities to personalized their engagement strategies to all alumni populations and adjust their approaches when necessary.

2.3 Design Principles for Alumni Management Platforms

While it is important to investigate the underlying reasons behind the market shift, it is also essential to address some of the fundamental design principles for a high-quality, successful alumni management platforms. Simplicity, consistency, user-centric, affordance and accessibility are all important guidelines to ensure that the application is intuitive, relevant and friendly for all users (Krug, 2000).

The principle of simplicity is grounded in the concept of minimalism. It states that an application should be minimalistic with only the very essential functions included. By increasing clarity, optimizing efficiency, and enhancing aesthetics(Tenner, 2015), simple and uncluttered design patterns can facilitate user's understanding of the app, and allow them to navigate the interface with ease. Developers can focus on the most important elements in the application and make them more prominent to the users, improving the clarity and readability of the interface. Without unnecessary clutters and distractions, users will be able to focus on what they need to accomplish while using the application. Therefore, simple design can also reduce the time it takes for users to complete tasks, and thus help increase efficiency. Lastly, the principle of simplicity can also make the application aesthetically agreeable. The choice and coordination of space, font, and color can reflect such design principle to create a visually appealing, modern, and sleek outlook.

The principle of consistency in design states that across the entire interface, it's important to keep same elements to function the same way. By keeping all elements consistent, it will help improve usability, prevent users from cognitive overload, enhance brand recognition, and saves development effort (Tidwell, 2010). Without too much mental effort and onboarding training, users typically can recognize and process consistent elements that look and function the same way, as they encounter them while navigating the interface. They can usually anticipate how certain elements will behave based on their previous experiences with similar elements on previous pages. Moreover, consistency in design elements such as color, font, and icon can establish a brand image and increase brand awareness. Users can quickly identify the brand by its consistent visual language. There are many successful products in the past that employed this specific strategy, leading to enormous financial success, iconomic brand identities and loyal customer base (Airey, 2009), such as Apple, Lexus, and Oral-B. Last but not least, the principle of consistency is an economical choice for developers. It is crucial to save developer's time and effort by reducing the need of creating separate design elements for every single feature they want to implement, especially given the amount of repetitions. Developers can reuse existing design patterns, styles, and components, which can speed up the development process and reduce costs.

The principle of user-centric design consists of a thorough user research, representative persona creation, and iterative user testing (Sauro and Lewis, 2016). It starts with user research to understand users' goals, behaviors, and preferences for the intended application. User research typically take places in the format of surveys and user interviews. Based on user research, developers usually create fictional personas that represent the target user group to capture and generalize their main characteristics. After persona has been used for the first prototype design, an application will be put into user testing, which involves observing users interacting with the interface to identify any usability issues or areas for improvement. User testing is an iterative process that involves constantly making changes based on trials and feedback. Developers will prioritize usability by ensuring that the interface is easy to navigate, understand, and use (Pu et al., 2011). By putting user's needs as a priority, the goal of employing a user-centric design approach is to increase user satisfaction and engagement, and ultimately, the success of the product or service.

The two principles, affordances and accessibility, go hand in hand in designing an application (Dahlström, 2019). Affordance refers to the user-perceived or actual ability of an object or element to perform a particular action, whether or not it is intended by the developers. Accessibility refers to the ability of an interface to be used by population with disabilities. Affordance measures how strong a cues or signal is to suggest how an element should be interacted with on an interface. For example, a raising button suggests that it can be clicked or tapped. It implies that users will be able to take actions once the button is triggered (Gaver, 1991). The principle of accessibility protects the interests of disadvantaged users with visual, auditory, motor, or cognitive impairments (McGookin et al., 2008), advocating for everyone to have equal access to the information and functionality provided by the interface. Accessibility can be achieved through various design techniques, such as providing alternative text for images, using high contrast colors, and providing keyboard shortcuts for users who cannot use a mouse (Friedman and Bryen, 2007). By providing clear and consistent affordances, developers can facilitate user interactions, including those with disabilities. By ensuring accessibility, developers will benefit all the users from the affordances provided.

CHAPTER 3
PROFILE SEARCH

The first step of this MA thesis is to collect alumni information, including the details of their education and work experiences. In the past decade, LinkedIn[1] has become a popular platform for professionals to showcase their skills and experiences. Considering its popularity and large user base, LinkedIn can be used as an effective medium to search and gather up-to-date information of alumni. However, considering the large amount of data we need to collect for each alumni, compared to scrape and then verify, it is more efficient to scrape all the necessary information after a candidate profile is successfully verified and matched with an alumni in the given alumni database. Therefore, in the initial step, I will only focus on collecting the links to alumni's LinkedIn profile. In order to automate the search process of profile links, mainstream APIs such as LinkedIn API and Google API, as well as web browser automation tool such as Selenium, are considered as potential options, each having their own advantages and disadvantages.

3.1 LinkedIn API

LinkedIn API[2] is an official API provided by LinkedIn that allows users to access LinkedIn data programmatically. If authorized, the developer's LinkedIn API can provide the information of user profiles on LinkedIn. It has a high volume of short-term and long-term quota that are generally enough for developers' use. However, LinkedIn API requires authentication and limits the search scope. The publicly-accessible standard LinkedIn API doesn't allow users to search beyond their own profile networks and further authorization is needed to acquire the other users' profiles.

[1] LinkedIn website: `https://www.linkedin.com/`

[2] LinkedIn API documentation: `https://developer.linkedin.com/`

3.2 Google API

Google API[3] is a collection of APIs provided by Google that allow users to access various Google services programmatically, including Google Search. Google Search API[4] is one of the best options among the candidate search algorithms. On the one hand, it renders search results quickly and accurately without bot detection issues. On the other hand, its daily and minute quota are not too low and can be easily circumvented by adding wait times between searches.

3.3 Selenium

Selenium[5] is a web browser automation tool that can simulate user interactions with web pages, allowing users to navigate through pages and extract data automatically [6]. In our case, Selenium can mimic manual LinkedIn profile searches on Google and render relatively reliable results. However, search engines like Google will detect traffic without user interactions and prevent automated searches by reCAPTCHA tests[7] that Selenium is unable to circumvent (Chapagain, 2019). ReCAPTCHA tests are security measures distinguishing between human users and automated bots on websites by requiring the user to perform certain tasks that are easy for humans but difficult for automated machines. A potential solution is to spread out the search process into evenly divided patches, and place a five-minute break between every two consecutive patches. This will address the problem, but simultaneously elongate the search, making it less efficient.

[3] Google API documentation: `https://developers.google.com/apis-explorer`

[4] Google Custom Search API documentation: `https://developers.google.com/custom-search/v1/introduction`

[5] Selenium developer guide: `https://www.selenium.dev/`

[6] Selenium Python documentation: `https://selenium-python.readthedocs.io/`

[7] ReCAPTCHA documentation: `https://www.google.com/recaptcha/about/`

3.4 Search Workflow

Considering Google API's optimal performance, I decide to apply Google's free Custom Search Json API for alumni's LinkedIn profile search. Specifically, the input search content for Custom Search Json API is $site : LinkedIn.com/in/$ and $name$, followed by $UniversityOfChicago$, which basically requests the API to return results for any LinkedIn profiles that matches with the alumnus's name and the keyword "University of Chicago". The implementation splits the alumni data into multiple batches for data collection to avoid potential internal errors caused by frequent search. The search process could return zero, one, or multiple matches. In my implementation, if there exists matched results, I only keep the first match. As a result, 2,986 out of 5,131 (58.2%) alumni's LinkedIn profile links are collected at this step.

CHAPTER 4

DATA VERIFICATION AND COLLECTION

After acquiring matched results from the search process, it's essential to verify whether the result is an exact match of the alumni on our record by comparing the API-returned information with the information in the ground-truth database, including name, school, degree, and program. Consequently, I verify each profile by extracting and checking web elements from the corresponding LinkedIn url. If the key information is successfully verified, I will then scrape the LinkedIn profile from the verified url. Specifically, for each profile, the verification process will check alumnus's full name, graduation school, graduation program and degree. If successfully verified, the algorithm will proceed to classify experiences into education or work, and scrape each type of experience from alumni profile. For education experience, the algorithm will obtain school name, duration, and description. For work experience, the algorithm will record company name, position, job type, location, and description. Because LinkedIn regularly updates its web elements and security measures to prevent bot activity on its platform, I designed multiple verification and scraping approaches to bypass these limitations. Each approach has been working for a while, but none of them is effective at this moment.

4.1 Verification using Web Elements

The first verification approach is to utilize the spacing in unordered lists (*ul tags*) embedding. It used to be possible to extract information from LinkedIn profiles by analyzing the spacing between the text in different elements of the *ul tags*. I notice the pattern that LinkedIn has 12 spaces for experience and 7 spaces for awards/certifications. However, LinkedIn discarded unordered list and replaced it with other elements. The spacings have thus become indistinguishable, and thus this approach failed to extract experience-related web elements.

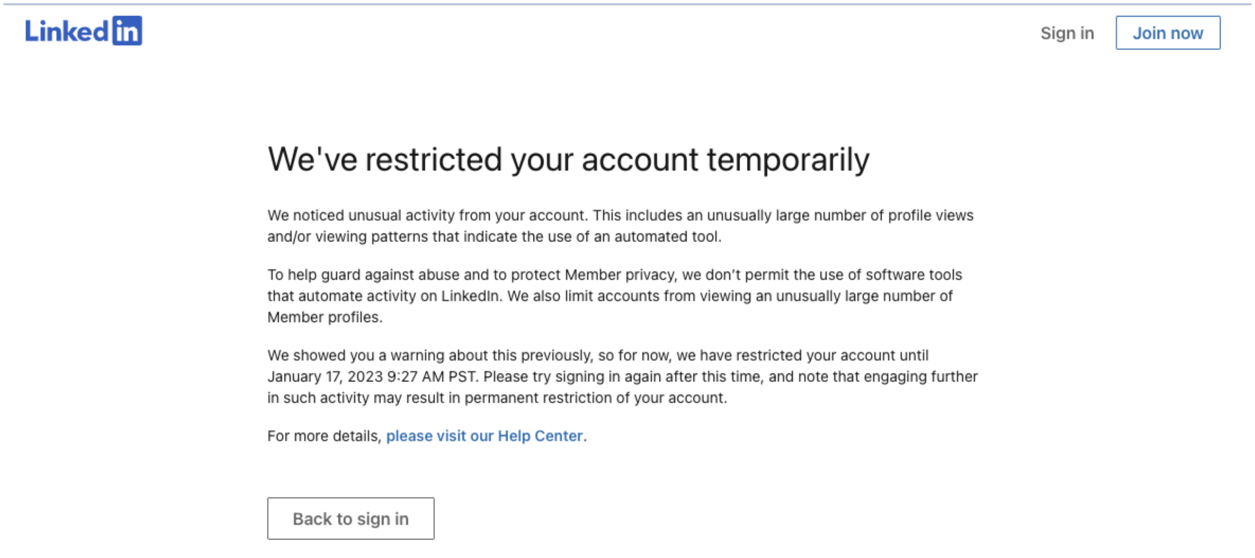
The second verification approach was to look for visually-hidden elements. After the deprecation of unordered lists in LinkedIn's web User Interface (UI), I implemented another verification approach that tracks down visually-hidden elements in the profile, which includes educational experience. However, LinkedIn then made various changes to its UI and web elements to prevent web scraping and automation. Now, the website has every element inside its own *CSS* and *div* classes to prevent functions like $find$ or $findall$ from the BeautifulSoup package[1]. As a result, this proposed approach was unable to search through and render text elements.

The last verification approach was based on *pvs-list container*. After the second verification approach fails, I resorted to scrape the giant *pvs-list container* element. The rationale was that it is relatively easy to clean up and categorize the experience after scraping, comparing to extract all list elements from the container, especially the $findall$ function from the BeautifulSoup package has stopped working. However, it soon turned out to be inoperable due to LinkedIn's restructure of its web elements.

4.2 Circumvent Bot Detection and Re-signins for Continuous Verifications

Two major issues this thesis has encountered for running verification algorithm in scale are: (1) how to prevent getting sign out by LinkedIn after around 30-50 searches and (2) how to avoid LinkedIn's bot detection mechanism. Automatic sign-outs cause interruption in the execution of verification program, and bot detection can lead to account restriction and suspension (e.g., Figure 4.1), or even illegal consequences. I tried multiple approaches to solve these issues.

[1] `https://beautiful-soup-4.readthedocs.io/en/latest/`



We've restricted your account temporarily

We noticed unusual activity from your account. This includes an unusually large number of profile views and/or viewing patterns that indicate the use of an automated tool.

To help guard against abuse and to protect Member privacy, we don't permit the use of software tools that automate activity on LinkedIn. We also limit accounts from viewing an unusually large number of Member profiles.

We showed you a warning about this previously, so for now, we have restricted your account until January 17, 2023 9:27 AM PST. Please try signing in again after this time, and note that engaging further in such activity may result in permanent restriction of your account.

For more details, please visit our Help Center.

Figure 4.1: Example LinkedIn Account Restriction Warning Screen

The first approach was to convert the collected alumni from a dataframe to a list of separate dataframes to loop from. After verifying each dataframe, I added a break to avoid LinkedIn's bot detection. However, LinkedIn still automatically sign out my account and I was unable to log back in while the program was executing.

To prevent interruption in program execution, I tried to automate sign out and sign into LinkedIn account at designated times while the verification algorithm is running. Nonetheless, The web elements necessary to sign out of LinkedIn were unable to be scraped or accessed programmatically, leading to another setback.

After that, I explored alternate browser options and added browser activities to bypass bot detection. I tried selenium in combination with firefox or chrome and included browser actions such as scrolling and clicking. However, such attempt turned out to be unsuccessful.

On top of mimicing sign-out and sign-in and alternative brower options, I also took a simple but less efficient way of verification and scraping. I created several LinkedIn accounts, alternate and only verify a few from each at one go. And when all of the previous solutions failed to provide effective verification on a large scale, it is natural to fall back to creating multiple fake LinkedIn accounts and only verifying a small number from each account. Although creating multiple fake LinkedIn accounts and alternating between them for verification is not the most efficient or cost-effective solution, it currently appears to be the only viable option for working around these limitations.

4.3 Verification Results

The verification and scraping algorithm is only able to get through roughly half of the alumni before the proposed approaches became defunct. 618 out of 2,986 (20.7%) candidate profiles from the collection are successfully verified in this process. Their LinkedIn profiles are scraped and stored for further data exploration. Table 4.1 summarizes the collected data.

Total Alumni	Alumni Data with LinkedIn Profiles	Alumni Data Verified/Collected
5,131	2,986 (58.2%)	618 (12%)

Table 4.1: Data Collection Summary

CHAPTER 5

DATA EXPLORATION AND ANALYSIS

With the raw LinkedIn profile data acquired from the previous step, I have conducted an exploratory data analysis to get insights into the distribution of alumni placement, industry, and location, which are critical for effective alumni management and networking.

5.1 Data Preprocessing

The scraped data includes a significant amount of irrelevant and redundant information beyond educational and work experience, such as skills and endorsements, volunteering experience, honors and awards, licenses, as well as recent posts and activities. Additionally, the data contains many newline characters, special characters, quotes, and random spacing, making it difficult to read, comprehend, parse and analyze. It also contain incomplete, inaccurate or inconsistent data, such as spelling errors, formatting issues, or missing values. Given how messy, disorganized, repetitive and confusing the raw alumni data is, data preprocessing is an essential first step of data exploration.

The first step is text cleaning. The meaningless trailing whitespaces and special characters are removed. Different combinations of newline characters and whitespaces are used as a separator for different experiences or details of one experience, depending on their length and the observed pattern during manual reading. At the end of this step, instead of storing all experience of each alumni as one observation, the data is now organized by person and experience, meaning that each experience now becomes its own individual unit of analysis. Text cleaning allows alumni data to have more comprensible content, more standardized format and more consistent presentation, increasing its readability and usability.

The next step is data organization. Depending on the content and attributes of each experience, an experience is grouped into categories of education or work. Education experience is organized into 3 attributes: school name, duration, and description. Work experience is organized into 5 attributes: company name, position, job type, location, and description.

From 618 verified alumni, 4,256 experience are collected and organized: 1,468 education-related and 2,788 work-related.

5.2 Preliminary Analysis

After preprocessing, I perform an analysis to gain insights into the distribution of alumni placement, industry, and location. This information is valuable in a number of ways for alumni management and networking. For a preliminary analysis, this thesis uses keyword extraction as its primary technique to classify industry and position types. Based on education experience, an analysis of PhD and MBA placements for alumni who sought more advanced educational opportunities after obtaining a MA degree from University of Chicago's Social Science is presented. Similarly, an overview of industry, position, and location distribution based on alumni's work experience is presented.

5.2.1 Post-graduate Education

On the one hand, 126 out of 618 verified alumni (i.e., $\sim 20\%$) chose to pursue a PhD degree. Their PhD domains and specializations are diverse, ranging from humanities, social sciences, public policy, STEM, and Business. The most popular field of interests are Political Science, Sociology, and Anthropology. As shown in Figure **??**, alumni's PhD institutions are relatively spread out. The most popular destination of SSD alumni is the University of Chicago (i.e., 21 alumni), suggesting some level of homophily. Other popular destinations include The University of Texas at Austin, Yale University, and Northwestern University.

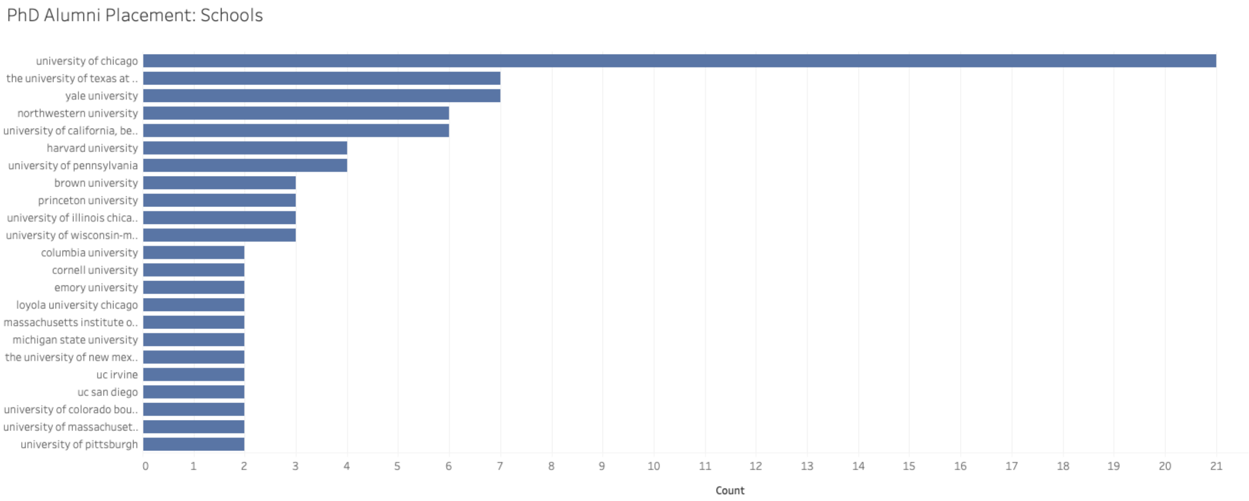


Figure 5.1: SSD Alumni PhD Placement: School

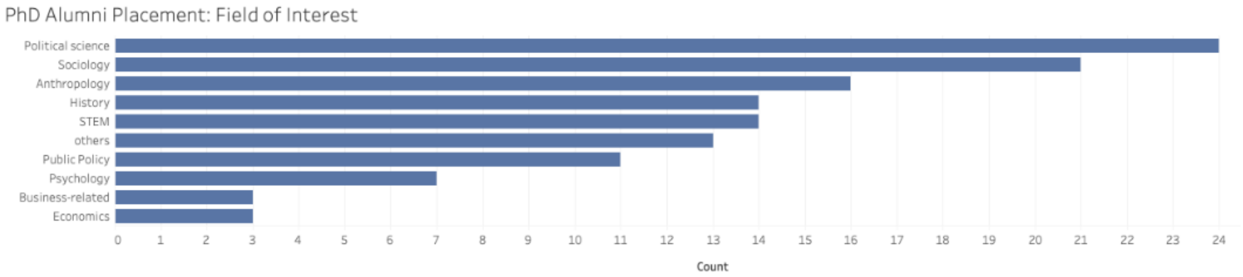
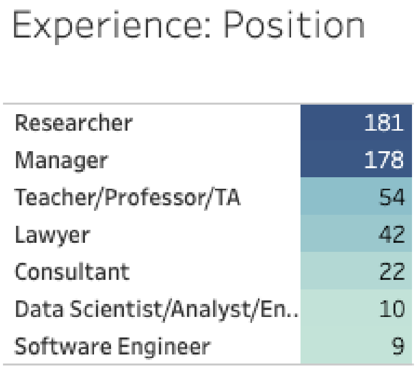


Figure 5.2: SSD Alumni PhD Placement: Field of Interest

On the other hand, 43 out of 618 verified alumni (i.e., $\sim$ 6%) chose to pursue an MBA degree. It was very impressive that a dominant majority of 31 out of 43 alumni decide to do their MBA at their alma mater, The University of Chicago's Booth School of Business, suggesting a higher level of institutional homophily compared to PhD placement.

5.2.2 Post-graduate Industry Work

Based on keyword extraction, each work experience is assigned a corresponding position type out of 7 pre-designed categories: Researchers, Manager, Teacher/Professor/TA, Lawyer, Consultant, Data Scientists/Analyst/Engineer, and Software Engineer (details in Table 5.3). The most popular position type among the SSD MA alumni are researchers and managers, each account for around 30% of alumni.



Experience: Position

Researcher	181
Manager	178
Teacher/Professor/TA	54
Lawyer	42
Consultant	22
Data Scientist/Analyst/En..	10
Software Engineer	9

Figure 5.3: SSD Alumni Job Placement: Position Type

Similarly, each work experience is assigned a corresponding industry type out of 10 pre-designed categories: Business/Finance, Charity/Volunteering, Consulting, Education/Research, Healthcare, Human Resources, Law, Policy/Government/Social Work, Technology and Others (details in Table 5.4). The most popular industry type is Education/Research and accounts for almost 30% of the job experience. Healthcare and Government are also popular industries among alumni. There are many experiences that are classified as “others", as they account for all different industry types other than the identified few. Otherwise, the distribution of other industry categories are fairly even.

In terms of location, SSD alumni can be found all over the world, with a national and global footprint that extends across the entire United States and even multiple continents (details in Table 5.4). The most popular location is the greater Chicago area, as some alumni stayed around the windy city after their graduation. In addition, there are three other popular metropolitan areas: Washington DC, New York City, and Boston. For alumni abroad, the most popular locations are Beijing, China, and Singapore.

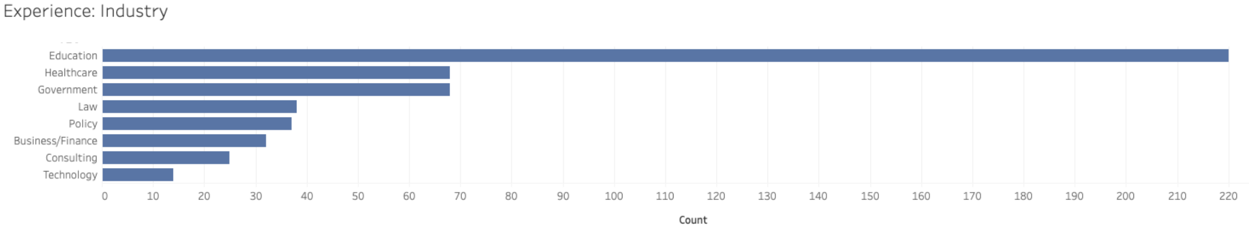
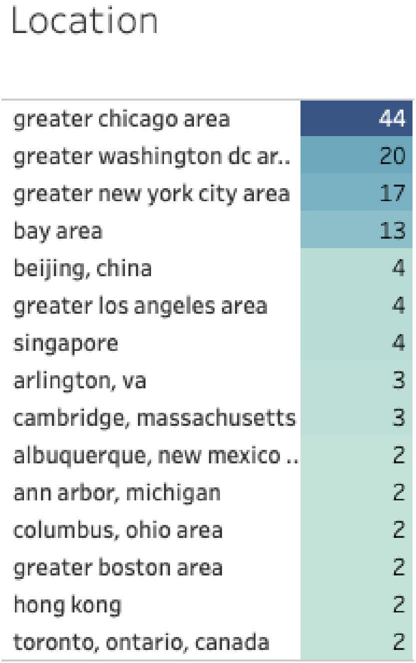


Figure 5.4: SSD Alumni Job Placement: Industry Type



Location

greater chicago area	44
greater washington dc ar..	20
greater new york city area	17
bay area	13
beijing, china	4
greater los angeles area	4
singapore	4
arlington, va	3
cambridge, massachusetts	3
albuquerque, new mexico ..	2
ann arbor, michigan	2
columbus, ohio area	2
greater boston area	2
hong kong	2
toronto, ontario, canada	2

Figure 5.5: SSD Alumni Job Placement: Location

5.3 Machine-Learning-Driven Industry Category Classification

In addition to an efficient and automated alumni database, this thesis also wishes to extend this initiative into a networking web app where alumni and current students can network and form mentorships based on career and research interests. A mentor matching algorithm can help identify potential matches between alumni and students based on their intended career. For the alumni data in this thesis, keyword matching is sufficient, but a more elaborate system should be implemented for large scale database. Therefore, this thesis expands beyond the original purpose of data collection to include machine learning-driven industry classifications. The previous keyword-extraction-based industry classification will be taken as ground-truth labels for the classification task, just for evaluation purposes. The process involves: (1) applying DistilBERT, an existing pre-trained language model, to classify the alumni work experience; (2) training a classification model with a larger dataset and then predict for the current alumni work experience data; (3) evaluating model performance.

5.4 Model Selection and Training

Alumni work experience dataset has 2,047 complete records. To choose the appropriate algorithm, I have considered several factors that influence the performance of models on similar classification tasks: size and similarity of the training and test datasets, the complexity of the classification task, and the quality of the features used in the model (Yang and Pedersen, 1997; Sun and Lim, 2001). Existing literature has shown that pre-trained models and techniques can help improve the performance of the model on sparse/small data from a different source but in a similar domain (Dodge et al., 2020; Guo et al., 2022; Shnarch et al., 2021). Considering the small, sparse, incomplete nature of the alumni work experience data, the benefit of applying pre-trained models is clear. However, training models from scratch helps to build more context specific classifiers and flexibility to fine tune parameters. Therefore, when designing the industry classification model and accounting for sparse data simultaneously, this thesis adopt two approaches: a pre-trained DistilBERT-based industry classifier, and a self-trained classifier.

The DistilBERT-based industry classifier[1] was trained and fine-tuned on a dataset of 7,000 samples of business descriptions from companies in India, which classifies these descriptions into one of 62 industry tags (John Snow Labs, 2021).

For the other self-trained classification model, I tried logistic regression and Naive Bayes, using a large dataset from Kaggle[2] with around 240,000 observations from job posts by companies from UK (Lee, 2018). This dataset is used to predict salaries for jobs based on a variety of attributes, including industry type, company, position, and job description. Nonetheless, it holds a very similar structure, includes all the feature variables and outcome variables necessary for the intended industry classification. In order to have the same industry classification tags as the smaller alumni work experience dataset for training purposes, I have created a mapping between the classification tags from this dataset and the tags in our own alumni work experience dataset. Most of these tags are not exact matches, but I have used my understanding of the industry landscape and the best judgement to assign the closest matching industry tag and ensure that the mapping is as accurate as possible to minimize any potential mis-classification errors in the final classification model.

The Kaggle dataset is split into training and testing for evaluation. I applied model in Python's scikit-learn package[3] to perform feature selection and ensure model generalizabil-

[1] John Snow Lab: DistilBERT Sequence Classification - Industry: https://nlp.johnsnowlabs.com/2021/11/21/distilbert_sequence_classifier_industry_en.html

[2] Text Analytics Explained-Job Description Data: https://www.kaggle.com/code/chadalee/text-analytics-explained-job-description-data/data

[3] https://scikit-learn.org/stable/

ity. Given the alumni work experience dataset is much smaller and sparser, identifying a subset of most predictive features can make the model more robust to new, unseen data, even when it's trained on a different dataset. Feature selection can also help to reduce the complexity of the model, making it faster and more efficient to train and evaluate on larger dataset. Following feature selection, the models are evaluated on test data within the larger Kaggle dataset based on accuracy, precision, recall and F1 score. Logistic regression marginally outperforms the Naive Bayes on the test set of the Kaggle dataset. It has an accuracy of 71% while the Naive Bayes classifier has an accuracy of 68%. This is in line with previous literature that has found Logistic Regression models to outperform other algorithms when it comes to simple text classification(Pranckevičius and Marcinkevičius, 2017). Eventually, the better-performing classifier, the logistic regression algorithm is selected and fit onto the alumni work experience dataset.

5.5 Results and Evaluations

Table 5.1 shows that Logistic Regression classifier does better with an accuracy of 22%, yet DistillBERT model has higher precision and recall. This is indicative that self-trained model on larger data is able to classify the right industries more accurately, but the pre-trained model has a higher quality of true positives or true industry labels, and has fewer false negatives across different industries.

Model	**Accuracy**	**Precision**	**Recall**
Logistic Regression	22%	11%	10%
DistillBERT	18%	15%	22%

Table 5.1: Model Performance on Alumni Work Experience Data

Job Category	**Precision**	**Recall**	**F1 Score**
Business	12%	4%	5%
Charity	5%	18%	8%
Consulting	4%	8%	6%
Education	25%	13%	17%
Healthcare	3%	2%	2%
HR	0%	0%	0%
Law	3%	1%	1%
Others	35%	50%	41%
Policy	3%	1%	2%
Technology	11%	10%	11%

Table 5.2: Self-Trained Logistic Regression Classifier Performance for Different Industry Categories

Job Category	**Precision**	**Recall**	**F1-Score**
Business	20%	32%	25%
Charity	14%	18%	16%
Consulting	0%	0%	0%
Education	20%	26%	23%
Healthcare	9%	65%	16%
HR	10%	41%	15%
Law	0%	0%	0%
Others	51%	4%	7%
Policy	4%	1%	2%
Technology	26%	32%	29%

Table 5.3: Pre-trained DistilBERT Classifier Performance for Different Industry Categories

Although the two models perform similarly in overall performance metrics, they exhibit differences in their performance when broken down by industry categories. On the one hand, the self-trained logistic regression classifier performs poorly on Human Resources and Healthcare categories, but relatively better on the Others and Education/Research categories. On the other hand, the pre-trained DistilBERT model performs relatively better in predicting the Business/Finance, Human Resources, Charity/Volunteering, and Technology categories, but the model performs poorly in predicting the Consulting jobs. Both models need to be improved on the Law category due to the missing classification tags for legal jobs in both training datasets.

5.6 Discussion

It is important to acknowledge that the conclusion about suboptimal performance of machine learning industry classification models entails several limitations, such as sparsity and homogeneity in alumni data, misrepresented ground truth, and various difference in training and test data. It should be interpreted with caution. Addressing these issues is essential for improving the generalizability of the findings from this thesis and ensuring that the models are reliable and effective in real-world settings. Therefore, in the last section, this thesis will address and elaborate on a few avenues that future developments can build on top of this thesis and make the findings from this thesis more robust.

5.6.1 Sparse Data and Homogeneity of Features

Due to data sparsity and homogeneity in feature variables, the findings from this thesis also have limited generalizability. Small and sparse test data in industry classification can be problematic because it leads to overfitting, inaccurate predictions, and difficulty in evaluating and improving the model. In this case, because the alumni dataset is so sparse, the poor performance of the models is likely caused by models overfitting to the training samples and not generalizing well to alumni data. Moreover, model evaluation and improvement is extremely difficult given the nature of the alumni data. If there are few or no occurrences of critical words or phrases in the test data, the models might have trouble classifying them properly. Thus, in model evaluation, it can be hard to distinguish between a good model that is simply making random errors and a good model that is systematically biased or flawed.

Moreover, the homogeneity of the feature variables, or in other words, the lack of numerical features across all the datasets presents significant challenges for the models. Text data is typically high-dimensional and sparse, meaning that there are many features with few occurrences or missing entirely, making it difficult for the models to accurately learn patterns and make predictions in the pre-training stage (Aggarwal and Zhai, 2012). In addition, text data is often noisy and has high variation (Veit et al., 2017), which can further complicate the modeling process.

5.6.2 Inaccurate Ground Truth

Classification is a supervised machine learning task. While evaluating the classifiers, true labels need to serve as benchmarks for predicted labels generated from new, unseen test data. By comparing the model's predicted labels to the true labels, we can compare, evaluate, and analyze the model performance. However, the ground truth from alumni work experience dataset is not from manual annotation, but instead generated based on keyword extraction and stays unverified. Since the accuracy of true labels is prone to errors, the conclusion from model evaluation might be better regarded as a comparison between industry classification using keyword extraction and machine learning approaches.

5.6.3 Differences between Train and Test Data

The pronounced differences between the training data and the alumni data from LinkedIn profile that could have potentially affected the model performance. One important distinction comes from job description. The self-trained classifier was pre-trained on descriptions posted by companies, but the job descriptions in alumni work experience dataset are self-reported. There might be some differences in the language, the intention and the level of detail provided which can influence model learning. Company-posted job descriptions are more formal, comprehensive, and higher-level. In addition to a description of the duties from the position, companies also tend to include requirements, qualification, salary range, and benefits, with the intention of filtering for ideal candidates and attracting a wide range of candidates. As opposed to company-posted job descriptions, self-reported job descriptions tend to be informal, personalized, and detail-oriented. They don't contain any qualifications and benefits, but usually include more details on the daily tasks and responsibilities. They often highlight the skills and accomplishments of an individual, with an intention to impress recruiters or potential employers.

In addition to job description, another important difference comes from from industry classification tags. The Kaggle dataset, the dataset DistilBERT model was pre-trained on, and the alumni work experience dataset don't have the same industry classification, and there is no exact mapping. The DistilBERT model was pre-trained on a dataset that has extensive subcategories from IT, Business, Finance, Healthcare, which facilitate mapping from these subcategories into the ten categories we designated. However, it doesn't have equivalent industry classification tags for Policy/Government/Social Work, Charity/Volunteering, and Law. It also lacks good coverage for Education/Research. Alternatively, the large Kaggle dataset has great coverage for jobs in Education and Research, but falls short on Human Resources, Technology, Healthcare, and Policy job categories.

The last set of differences includes geographical differences between the datasets which could affect model performance. The pre-trained DistilBERT model was trained on data from Indian companies, while the large Kaggle dataset was drawn from UK company postings. The alumni dataset, on the other hand, includes self-reported experiences from alumni all over the world, with profiles written in different languages and translated into English. As a result, owing to differences in language use, cultural context, and reality in local job markets, the pre-trained DistilBERT model might perform better on Indian alumni, and the self-trained model might perform better on British alumni.

CHAPTER 6

SSD CONNECT: AN ALL-IN-ONE ALUMNI INFORMATION MANAGEMENT SYSTEM (PROTOTYPE)

From the user interviews I have conducted, students, alumni and administrators from SSD have openly expressed their expectation towards integrating the existing networking platforms: MA Connect, Grad Gargoyle, LinkedIn, and the program-specific mailing lists. Lacking an all-in-one, consolidated platform for all networking needs or missing features targeting program-specific and personalized networking needs, the current networking structure for alumni and students is very decentralized, confusing, and difficult to navigate. On top of that, it is not cost-efficient from the University's perspective. Both MA Connect and Grad Gargoyle require expensive purchase and updates, and the administrators have to put many hours into maintenance and often need to post overlapping information on these platforms. Therefore, besides a relevant, efficient, automated alumni database, this thesis also extends the initiative of creating an integrated networking web app that alumni and current students can network and form mentorships based on career and research interests.

This prototype is a front-end interface that consolidates the functions of LinkedIn, MA Connect, Grad Gargoyle, and email lists. There are three types of roles: administrator, current student, and alumni. All users share some common features, such as establishing networks, signing up for events, monitoring chats, and receiving notification. Current students will have opportunities to request and form mentorships based on matching requests, and report offers through the app. Administrators will have management features such as posting events, getting real-time analytics of user groups, as well as adding and removing users. Figure 6.1, 6.2, and 6.3 show an overview of the corresponding interfaces, and the link to a video walk-through of the complete prototype is available here: `https://drive.google.com/file/d/1So3Dup-UzgJ01v_fsDZ0IYkb2Hx2EMaE/view?usp=sharing`

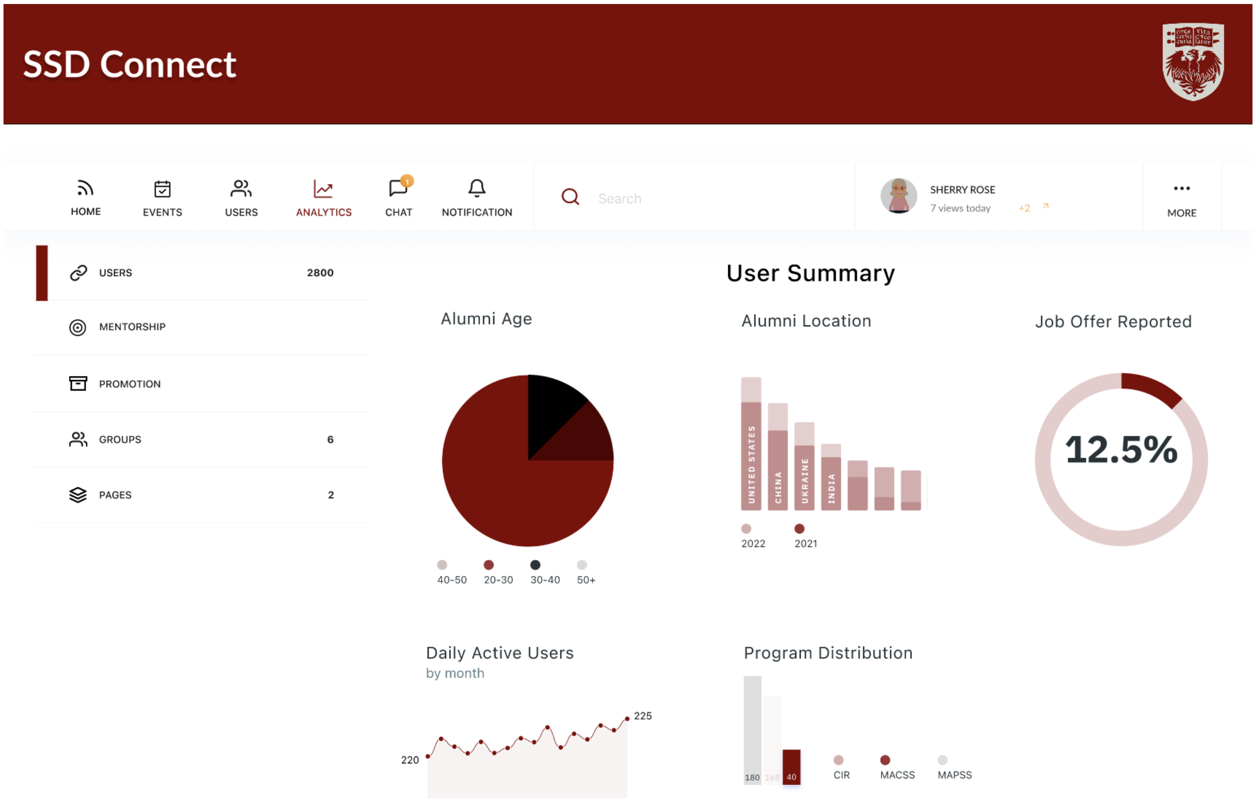
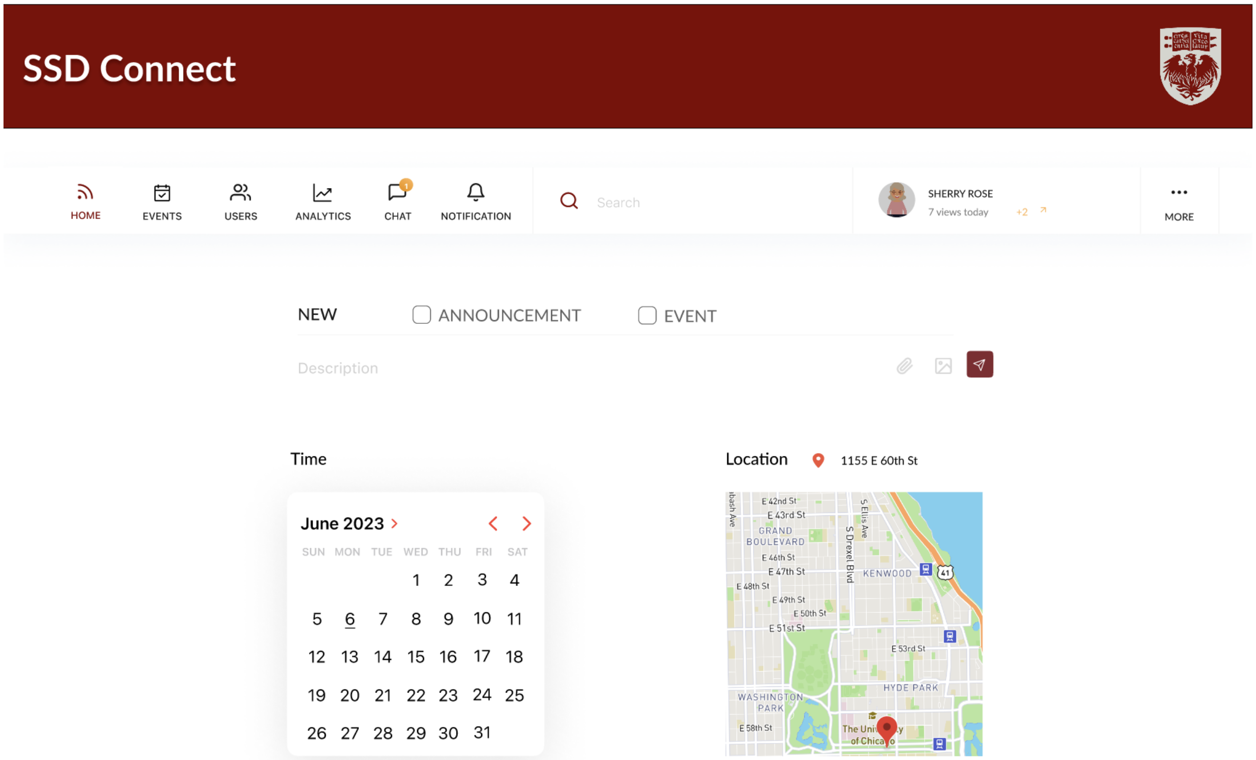


Figure 6.1: SSD Connect: Administrator Analytics Feature

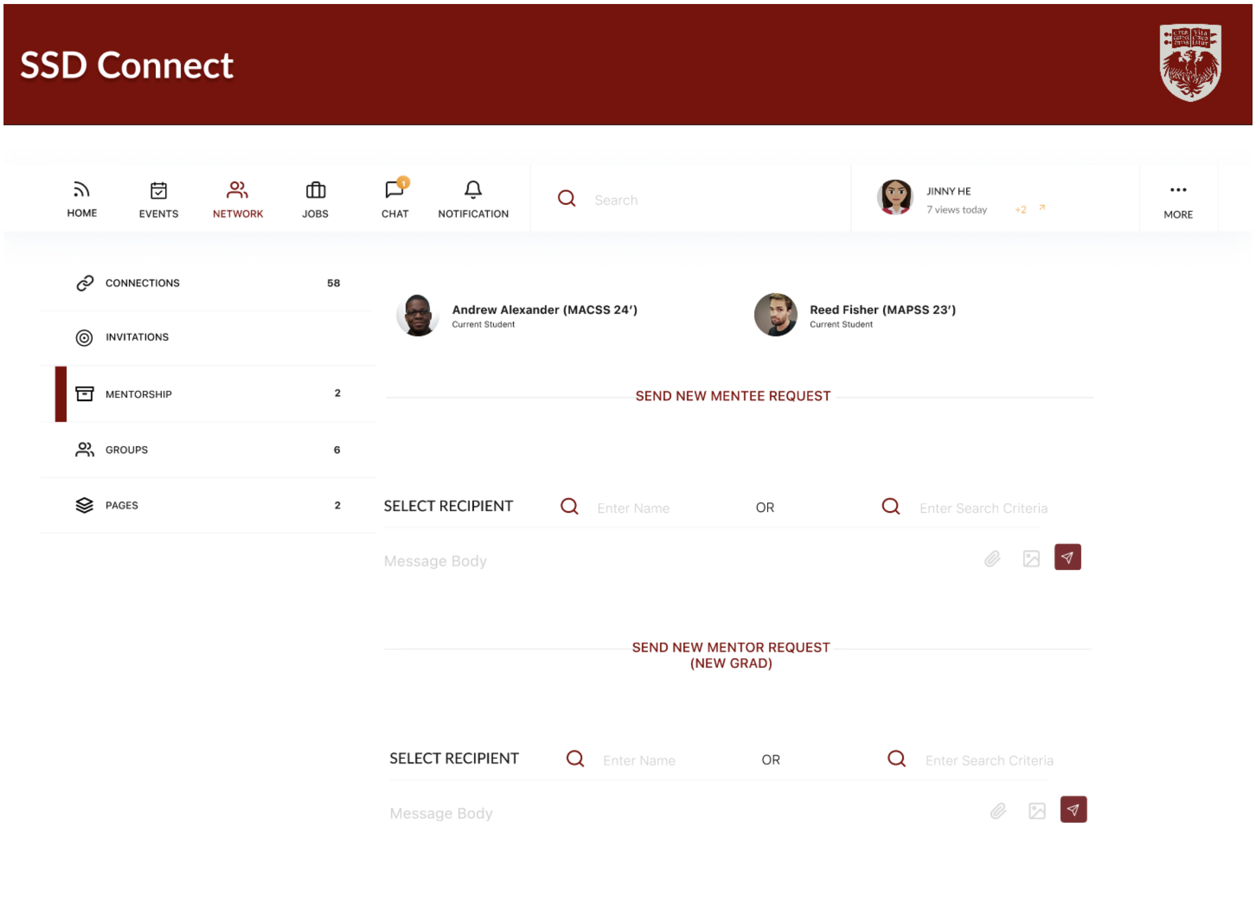


Location 1155 E 60th St

June 2023

SUN	MON	TUE	WED	THU	FRI	SAT
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Figure 6.2: SSD Connect: Administrator Event Feature



CONNECTIONS	58
INVITATIONS	
MENTORSHIP	2
GROUPS	6
PAGES	2

Andrew Alexander (MACSS 24′)
Current Student

Reed Fisher (MAPSS 23′)
Current Student

Figure 6.3: SSD Connect: Alumni Mentorship Feature

CHAPTER 7

CONCLUSION

Built on previous literature, this thesis explores the possibilities to maximally automate alumni information collection for SSD, highlights the setbacks due to LinkedIn's changing web structure and restrictions on data scraping, yet proposes solutions to overcome these challenges. The preliminary analysis provides a breakdown of SSD alumni's PhD and MBA placements, as well as their work experience in terms of position, industry, and location, allowing different stakeholders to gain insights into the career trajectories and employment outcomes of SSD alumni.

On top of that, this thesis uncovers the reasons behind the success story of existing professional alumni management and networking platforms and proposes different industry classification models using both keyword extraction and machine learning methods for mentorship matching. Compared to keyword extraction, the machine learning models render very different industry classification results, suggesting a significant potential for using machine learning techniques to improve the accuracy and effectiveness of industry classification in alumni networking and mentorship matching. Nevertheless the findings have prominent limitations and should be interpreted with caution.

Last but not least, a prototype of an all-in-one alumni management and networking web application is created as an extension of this thesis and provides a foundation for future development, integration, and improvement of SSD's career service and alumni outreach program.

Overall, the thesis contributes to the body of literature on alumni networks by brining in novel methods and insights for managing and analyzing these networks. The carefullydesigned automated information collection pipeline is replicable, and the discussion on proposed mentor matching industry classification algorithms can be very useful for future alumni management and networking tool. These innovations and developments can potentially help organizations and institutions like SSD to improve the alumni management strategies, and serve as a great resource for future research and development in several academic and industry topics, including alumni outreach, career services, natural language processing, text classification, machine learning, web development, and user interface/user experience design.

CHAPTER 8

FUTURE WORK

8.1 Data Collection and Verification

Future researchers should look into LinkedIn website, investigate how LinkedIn restructures its web elements, identify the constants and scrapable elements to develop a better, more efficient, and most importantly durable and reliant scraping algorithm against new updates without violating any terms of service, or getting temporary and even permanent account suspensions. This can improve the accuracy and reliability of data source for alumni placement analysis and mentor matching algorithm, as well as minimize the risk of legal or ethical issues from LinkedIn scraping.

8.2 Industry Classification Model

There are a few directions future research can proceed with the design of industry classification machine learning models. In terms of model design, future work should consider other pre-trained models that are trained both within and outside of the job description and industry classification context, and experiment with test and sentence embeddings for classification. It will also be beneficial if future research adopt a more rigorous feature selection process to train and evaluate models from scratch to boost results.

In terms of data, future work could explore other potential data sources that augment the data for our model for better performance.

Another potential strategy could be a combination of predictions from both models. In particular, one could compare the results from each model and select the prediction from the model with the better performance on the given industry category. For instance, it might be a good idea to try using the pre-trained DistilBERT model for IT, Business/Finance, and healthcare job postings and the self-trained model for other job categories. This approach could improve the overall accuracy and reliability of the task.

Lastly, to make these NLP models and algorithms less blackboxed, future work can undertake error analysis to better understand why models get what they get right, and where they perform poorly. This would be meaningful not only for an extension or our research, but the larger NLP community.

8.3 SSD Connect Prototype

The front end prototype for SSD Connect will be a great reference for future web developers and UI/UX designers. There are three potential paths: elaborate and refine the front end, move to the backend and complete the integration of both front- and back-end of the web application, or develop a mobile version.

To improve front-end features, visuals, or structure, user research to evaluate the usability and effectiveness of the web application for intended audience can be a good way to start. While this thesis did informal user interviews and surveys for prototype design, the scale was very small. Further research can recruit more participants from a diverse background to gather feedback, identify areas for improvement, and improve front-end design.

To develop a strong, efficient back-end and integrate back-end into the application, it is necessary to have a secure, scalable, customizable database to store alumni information and design an efficient way of database update. It also needs to include modules for managing events, updating job board, sending communications, tracking offers, and generating analytics. The goal is to allow the system to integrate the back-end, interact with the front-end, and offer a perfectly-smooth user experience.

Last but not least, a mobile version, as discussed in literature review, is an important component for many successful professional alumni management and networking services. A mobile version alongside with a web application will be the best for user experience and is recommended for alumni management and networking tools.

CHAPTER 9

SUPPLEMENTAL MATERIAL

Replication data for this project can be found on a GitHub repository contains all the data verification, collection, analysis, and visualization of our data, and allows for the replication of all tables and figures included in this paper:
https://github.com/MACSS-Projects/Lynette-thesis-alumni-2022

The prototype for SSD Connect is published on Figma Community for sharing purpose:
https://www.figma.com/community/file/1216973904264301889/SSD-Connect-Prototype

REFERENCES

Aku Aarva and Pauli Alijärvi. Creating a foundation for international alumni networks in the finnish universities of applied sciences: Case jamk university of applied sciences. 2012.

Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. *Mining text data*, pages 163–222, 2012.

Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

David Airey. *Logo design love: A guide to creating iconic brand identities*. New Riders, 2009.

Daniela Aliberti, Rita Bissola, Barbara Imperatori, Francesca Mochi, et al. Growing brand ambassadors: The role of affective commitment, person-organization fit and networking behaviors in the context of alumni networks. *IMPRESA PROGETTO*, 2022(1):1–18, 2022.

Mohamed Almorsy, John Grundy, and Ingo Müller. An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*, 2016.

Philip G Altbach and Jane Knight. The internationalization of higher education: Motivations and realities. *Journal of studies in international education*, 11(3-4):290–305, 2007.

American Psychological Association. *Publications Manual*. American Psychological Association, Washington, DC, 1983.

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, December 2005. ISSN 1532-4435.

Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, 2007.

Thibaut Bardon, Emmanuel Josserand, and Florence Villesèche. Beyond nostalgia: Identity work in corporate alumni networks. *Human Relations*, 68(4):583–606, 2015.

Badan Barman. Learning alumni management from the top ten ranking universities in nirf-2019 and its application in developing a custom social network for management of alumni of a department of library and information science. *Learning*, 1:3–2019, 2019.

Zenia Barnard. *Online community portals for enhanced alumni networking*. PhD thesis, University of Johannesburg, 2007.

Zenia Barnard and Chris Rensleigh. Investigating online community portals for enhanced alumni networking. *The Electronic Library*, 26(4):433–445, 2008.

Benjamin Borschinger and Mark Johnson. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia, December 2011.

Morgan Brattstrom and Patricia Morreale. Scalable agentless cloud network monitoring. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 171–176. IEEE, 2017.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981. doi:10.1145/322234.322243.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '20, page 3163–3171, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi:10.1145/3394486.3403368. URL `https://doi.org/10.1145/3394486.3403368`.

Anish Chapagain. *Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others*. Packt Publishing Ltd, 2019.

Tzuo-Ming Chen and Chi-Tsai Yeh. Integrating facebook and alumni into the social network mobile platform. *International Journal of Electronic Commerce Studies*, 5(1):109–114, 2014.

Hongmei Chi, Edward L Jones, and Lakshmi P Grandham. Enhancing mentoring between alumni and students via smart alumni system. *Procedia Computer Science*, 9:1390–1399, 2012.

David Chiavacci. Transition from university to work under transformation: the changing role of institutional and alumni networks in contemporary japan. *Social Science Japan Journal*, 8(1):19–41, 2005.

Helene Dahlström. Digital writing tools from the student perspective: Access, affordances, and agency. *Education and Information Technologies*, 24:1563–1581, 2019.

Debao Dai and Yusen Lan. The alumni information management model based on" internet+". In *7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)*, pages 38–42. Atlantis Press, 2017.

Alexandra David and Frans Coenen. Alumni networks–" an untapped potential to gain and retain highly-skilled workers?". *Higher education studies*, 4(5):1–17, 2014.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Karin Ebert, Leona Axelsson, and Jon Harbor. Opportunities and challenges for building alumni networks in sweden: A case study of stockholm university. *Journal of Higher Education Policy and Management*, 37(2):252–262, 2015.

Association for Computing Machinery. *Computing Reviews*, 24(11):503–512, 1983.

Mark G Friedman and Diane Nelson Bryen. Web accessibility design recommendations for people with cognitive disabilities. *Technology and disability*, 19(4):205–212, 2007.

Maria L Gallo. How are graduates and alumni featured in university strategic plans? lessons from ireland. *Perspectives: Policy and Practice in Higher Education*, 22(3):92–97, 2018.

William W Gaver. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 79–84, 1991.

James Goodman, Andreas Vlachos, and Jason Naradowsky. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11. Association for Computational Linguistics, 2016. doi:10.18653/v1/P16-1001. URL `http://aclweb.org/anthology/P16-1001`.

Biyang Guo, Songqiao Han, and Hailiang Huang. Selective text augmentation with word roles for low-resource text classification. *arXiv preprint arXiv:2209.01560*, 2022.

Dan Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK, 1997.

Sarah Hall. Educational ties, social capital and the translocal (re) production of mba alumni networks. *Global Networks*, 11(1):118–138, 2011.

Mary Harper. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1. Dublin City University and Association for Computational Linguistics, 2014. URL `http://aclweb.org/anthology/C14-1001`.

John Snow Labs. Spark nlp. Retrieved July 11, 2021, from `https://nlp.johnsnowlabs.com`, 2021.

Steve Krug. *Don't make me think!: a common sense approach to Web usability*. Pearson Education India, 2000.

Chad Lee. Text analytics explained: Job description data. Kaggle, 2018. URL `https://www.kaggle.com/code/chadalee/text-analytics-explained-job-description-data/data`.

David McGookin, Stephen Brewster, and WeiWei Jiang. Investigating touchscreen accessibility for people with visual impairments. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, pages 298–307, 2008.

Aritra Mukherjee, Adrita Roy, Manish Kumar Lath, Arnab Ghosal, and Diganta Sengupta. Centralized alumni management system (cams)-a prototype proposal. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 967–971. IEEE, 2019.

Kanal Paul Nigam. *Using unlabeled data to improve text classification.* Carnegie Mellon University, 2001.

Rabi Prasad Padhy, Manas Ranjan Patra, and Suresh Chandra Satapathy. Cloud computing: security issues and research challenges. *International Journal of Computer Science and Information Technology & Security (IJCSITS)*, 1(2):136–146, 2011.

Adeel Patras. International students alumni network. 2020.

Tomas Pranckevičius and Virginijus Marcinkevičius. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, 5(2):221, 2017.

Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164, 2011.

Mohammad Sadegh Rasooli and Joel R. Tetreault. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733, 2015. URL `http://arxiv.org/abs/1503.06733`. version 2.

İsa Sağbaş, Naci Tolga Saruc, and Çiğdem BÖRKE TUNALI. How do universities contact their alumni? practices of the best universities in the world university rankings. *Yükseköğretim Dergisi*, 8(3):334–345, 2018.

Jeff Sauro and James R Lewis. *Quantifying the user experience: Practical statistics for user research.* Morgan Kaufmann, 2016.

Selenium with Python Contributors. Selenium with python documentation. Read the Docs, 2023. URL `https://selenium-python.readthedocs.io/`.

Johanna Shih. Circumventing discrimination: Gender and ethnic strategies in silicon valley. *Gender & Society*, 20(2):177–206, 2006.

Eyal Shnarch, Ariel Gera, Alon Halfon, Lena Dankin, Leshem Choshen, Ranit Aharonov, and Noam Slonim. Cluster & tune: Enhance bert performance in low resource text classification. 2021.

Amber L Stephenson and David B Yerger. The role of satisfaction in alumni perceptions and supportive behaviors. *Services Marketing Quarterly*, 36(4):299–316, 2015.

Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528. IEEE, 2001.

Gislaine Cristina dos Santos Teixeira and Emerson Antonio Maccari. Proposition of an alumni portal based on benchmarking and innovative process. *JISTEM-Journal of Information Systems and Technology Management*, 11:591–610, 2014.

Edward Tenner. The design of everyday things by donald norman. *Technology and Culture*, 56(3):785–787, 2015.

Jenifer Tidwell. *Designing interfaces: Patterns for effective interaction design.* " O'Reilly Media, Inc.", 2010.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847, 2017.

Fredrick Hoopes Wampler. *Bridges to a lifelong connection: A study of Ivy Plus young alumni programs designed to transition recent graduates into engaged alumni.* University of Pennsylvania, 2013.

Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35. Citeseer, 1997.

Ziyi Zhang, Shuofei Zhu, Jaron Mink, Aiping Xiong, Linhai Song, and Gang Wang. Beyond bot detection: Combating fraudulent online survey takers. In *Proceedings of the ACM Web Conference 2022*, pages 699–709, 2022.