Improving Quality of Job Application Pre-Processing with Knowledge Graphs

by
Joseph Porter, Jr.

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

at

Seidenberg School of Computer Science and Information Systems

Pace University

December 2020

We hereby certify that this dissertation, submitted by Joseph Porter, Jr., satisfies the dissertation requirements for the degree of *Doctor of Professional Studies in Computing* and has been approved.

_______________________________________ _______________

Dr. Lixin Tao Date: December 15, 2020

Chairperson of Dissertation Committee

_______________________________________ _______________

Dr. Charles Tappert Date: December 15, 2020

Dissertation Committee Member

_______________________________________ _______________

Dr. Tilak Agerwala Date: December 15, 2020

Dissertation Committee Member

Seidenberg School of Computer Science and Information Systems

Pace University

We hereby certify that this dissertation, submitted by Joseph Porter, Jr., satisfies the dissertation requirements for the degree of *Doctor of Professional Studies in Computing* and has been approved.

Lixin Tao

Dr. Lixin Tao
Chairperson of Dissertation Committee

Date: December 15, 2020

Charles Tappert
Charles Tappert (Dec 16, 2020 22:22 EST)

Dr. Charles Tappert
Dissertation Committee Member

Date: December 15, 2020

Tilak Agerwala
Tilak Agerwala (Dec 18, 2020 13:01 EST)

Dr. Tilak Agerwala
Dissertation Committee Member

Date: December 15, 2020

Seidenberg School of Computer Science and Information Systems
Pace University

Abstract

Improving Quality of Job Application Pre-Processing with Knowledge Graphs

by
Joseph Porter, Jr.

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

December 2020

Human Resources (HR) personnel face challenges when preprocessing or evaluating job applications. The main goal of job application preprocessing is to filter through many resumes in order to produce a short list of the most qualified candidates to hire. The lack of domain expertise in high-tech recruiting has led to poorly qualified prospective employees advancing to the short list of candidates considered for hire. The challenge in evaluating prospective employees is made worst given the variety of professional and laymen terms used by job-providers/companies and job-seekers/prospective employees. Job description created by companies and resumes created by prospective employees both use a combination of professional and laymen terms to describes job requirements and job qualification. It is acceptable for one company to use different terms than another to describe a Cloud Software Developer's position. Similarly, a prospective employee can use different terms than another perspective employee in describing qualifications for a Cloud Software Developer's position.

Ontology is often used to define vocabulary and terms in an application domain. However, the most popular ontology tools in used today are limited in supporting only the single "is-a" relationship which prevent them from describing rich relationship needed to capture terms used during job application evaluation or preprocessing. To fill this gap, Pace Universities Knowledge Graph (KGs) and manual methods of producing KGs extend ontology to support customized relationship like "part of" to better support relationships among various concepts that may be used by professional and laymen in job descriptions or resumes.

This research proposes to use KGs to identify custom relationship so that important keywords could be used in a Python Analyzer (PA) Application Tracking System (ATS) that was developed for matching keywords in job descriptions to keywords in resumes. The PA ATS use KG keywords from a text file to compare against job description keywords to produce a subset of keywords for further matching between the job description and a batch of resumes for the job. For each resume, HR personnel are able to see match percentage before and after filters for keyword and keywords with synonyms were applied. This research experiments showed a 10% average improvement in job match accuracy using the PA ATS. Improved matches enhance job application preprocessing.

Contribution in this research can also be applied to similar problems that need a PA ATS tool to bridge terminologies in different communities.

Acknowledgements

I would like to acknowledge Dr. Lixin Tao (as well as the entire faculty and support staff at *Pace University*) for making this Doctor of Professional Studies in Computing program possible. I would also like to acknowledge Dr. Charles Tappert, and Dr. Talik Agerwala.

Special thanks are extended to DPS students and faculty who offered support and encouragement, as well as who provided ideas and food for thought. Communicating with everyone helped me to stay focused.

I also want to thank Team-5 which includes Dave, Edmund, Binu and Raj. My special thanks to Suzanna Schmeelk, Stephan Barabasi and David Lasecki. Their support with the Python Analyzer and encouragement was essential to the completion of this work. Thank you everyone for your help in allowing me to gather information for this dissertation.

A special acknowledgement goes to my wife, Sherri whose endless support and love made this all possible. And, to our children—Joseph and Eric—I love you so much.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Resume Processing Pitfall

Resume processing technology has a tough time matching job seekers with job providers because of differences in words or terms used in job description versus words and terms used in resumes. Miss-matches due to synonyms are especially problematic. Companies tend to include formal, professional words in job descriptions that differ from less formal lay terms that may be included in resumes. In January 2019, CareerBuilder's CEO. Irina Novoselsky made this comment:

"In our nearly 25 years of experience, we've learned that recruiters speak one language and candidates speak another. We are building and implementing technologies that close this gap for both sides of our marketplace, and improve every stage of the hiring process, …" [18]

Irina's comments indicates that there are pitfalls when it comes to using keywords to match a job provider with a job seeker. Below, I highlight pitfalls and challenges that result in lower match percentages due to an Applicant Tracking System (ATS) or other software that does not include Knowledge Graphs keywords and synonyms (KGKWS) to help find matches.

Most companies use keywords found in ATSs to find matches between resumes and job descriptions. Keyword matching alone is not the best because this type of match has trouble recognizing synonyms or words that have the same meaning. "Keyword matching is prone to inaccuracies because it doesn't take into account things like your skill set and qualifications. You could search on keywords and get a lot of irrelevant results, while at the same time missing out on results that would actually be well-suited to your particular skills mix and qualifications." [14]

Sections 1.1.1, 1.1.2 and 1.1.3 below describe use-cases which highlight problems associated with blackbox keyword (BBKW) matching between resumes and job descriptions.

1.1.1 Recruiter's Challenge Resolving no match on "Excel" when Microsoft Office is listed Use Case

Table 1 Recruiter's Challenge Resolving no match on "Excel" when Microsoft Office is listed Use CaseTable 1

Use Case Name	Recruiter's Challenge **Resolving no Matches** on "Excel" when Microsoft Office is listed Use Case
Primary Actor(s)	Recruiter
Description	The Recruiter's use case for identifying a no match situation when matching algorithm does not know that Excel is a part of Microsoft Office.
Precondition	Recruiter must have a resume and a job description to feed into an Applicant Tracking System (ATS) which compares words in both documents.
Post condition	The ATS must produce a report for the Recruiter based on a comparison and match of keywords and synonyms found in the two documents. The synonym matching portion does not exist currently.
Basic Course of Action	
Recruiter must run pre-processor program and get to entry screen	

Recruiter can then cut and paste a resume in the resume section of the screen and a job description in the job description section of the screen (or run a batch program that does this). Recruiter can click on the submit button to compare words in the two documents. Recruiter can then click on the "View Report" button to see words that matched as synonyms. End of use case.
Alternative Course of Action

1.1.2 Recruiter's Challenge Resolving Coder Versus Programmer Synonym Matching Use Case

Table 2 Recruiter's Challenge Resolving Coder Versus Programmer Synonym Matching Use CaseTable 2

Use Case Name	Recruiter's Challenge **Resolving Coder** versus Programmer Synonym Matching Use Case
Primary Actor(s)	Recruiter
Description	The Recruiter's use case for identifying that Coder is synonymous with Programmer
Precondition	Recruiter must have a resume and a job description to feed into an Applicant Tracking System (ATS) which compares words in both documents.
Post condition	The ATS must produce a report for the Recruiter based on a comparison and match of keywords and synonyms found in the two documents. The synonym matching portion does not exist currently.
Basic Course of Action	
Recruiter must run pre-processor program and get to entry screen Recruiter can then cut in paste a resume in the resume section of the screen and a job description in the job description section of the screen (or run a batch program that does this). Recruiter can click on the submit button to compare words in the two documents. Recruiter can then click on the button to view a report that shows words that matched as synonyms. End of use case.	
Alternative Course of Action	

1.1.3 Recruiter's Challenge Recognizing Synonyms for Cloud Matching Use Case

Table 3 Recruiter's Challenge Recognizing Synonyms for Cloud Matching Use CaseTable 3

Use Case Name	Recruiter's Challenge Synonyms for Cloud Matching Use Case
Primary Actor(s)	Recruiter
Description	The Recruiter's use case for identifying cloud software acronyms as synonym for cloud
Precondition	Recruiter must have a resume and a job description to feed into an Applicant Tracking System (ATS) which compares words in both documents. GCP = Google Cloud Platform EC2 = Elastic Compute Cloud (EC2)
Post condition	The ATS must produce a report for the Recruiter based on a comparison and match of keywords and synonyms found in the two documents. The synonym matching portion does not exist currently.
Basic Course of Action	
Recruiter must run pre-processor program and get to entry screen Recruiter can then cut in paste a resume in the resume section of the screen and a job description in the job description section of the screen (or run a batch program that does this). Recruiter can click on the submit button to compare words in the two documents. Recruiter can then click on the button to view a report that shows words that matched as synonyms. End of use case.	
Alternative Course of Action	

1.2 Problem Statement

This dissertation research identifies problems that recruiters face when attempting to select the best candidate to interview for a job. Below is a subset of problems that can be mitigated as a result of involving visual aids like KGs and adjusting matching algorithms to include synonyms and other filters.

1) **Problem One –** Recruiters spend lot of time manually filtering through resumes or pre-processing resumes trying to identify the best candidate from a pool of resumes to

recommend to hiring managers for an interview. And the short list of candidates is often not optimized.

2) **Problem Two –** ATS software that recruiters use to identify candidates rely on keyword matching which does not make allowances for synonyms and other important relationships. Instead of using keywords alone, "… it is important to consider whether a more general, or specialized keyphrase should be used, or whether it should be used with some synonym, or even replaced by another keyphrase." [16] Using keywords alone results in less optimal match rates between job provider and job seeker. For example:

I. a resume that uses "coder" as a term for "programmer" will result in a mismatch based on keywords without factoring in synonyms.

II. "server" may not match with "servers"

III. "wood-worker" may be synonymous with "carpenter"

IV. Research indicates that the following may not be recognized as synonyms: "'Relationship Manager's Assistant' = 'Junior Relationship Manager'= 'Junior Client Relationship Manager' = 'Junior Customer Relation Manager' = 'CRM Assistant' = 'Assistant Relationship Manager' = 'Assistant RM' " [9]

V. Trying to find a match on "medical assistant" could result in a hit on "assistant" positions that have little to do with medicine as well as a hit on "medical" that has little to do with and assistant.

VI. Nonprofit may not match non-profit

VII. Three years of experience may not match 3 years of experience

VIII. CRM software will not match Saleforce or Siebel

IX. MS or MBA may not match Masters of Science or Master of Business Administration

3) **Problem Three –** Recruiters are often not experts in the STEM fields for which they recruit. As a result, recruiters rely on ATS software to sort through resumes looking for keywords that are important for the recruiting area in which they represent.

4) **Problem Four –** Companies use different terms in job descriptions to describe the same job, and job seekers often used word that are synonymous with word in job descriptions, but are not recognized as such and leads to mis-matches. Below are job descriptions that essentially the same, but are described differently by the 3 companies that list jobs on their websites:

I. System Software Engineer, Google Cloud Platform (GCP)

II. System Development Engineer - Elastic Compute Cloud (EC2)

III. Software Developer – IBM Cloud Hosting Team

5) **Problem Five –** Ontology tools that can help visualized relationships among keywords are limited in the kinds of relations that can be depicted. Most tools support only "is-a" relationships versus other rich relationships like "partOf."

1.3 Improving Resume Processing Accuracy with Knowledge Graphs (informal ideas, use cases)

As pointed out earlier, many pitfalls associated with software keyword matching can be eliminated, improved or reduced through the use of Knowledge Graphs (KGs). KGs allow visual relationships to be created which helps recruiters to better understand relationship between keywords. For example, the Pace University KG system supports customized relationship between keywords so that relationships like the one in Figure 1 and Figure 2 are possible.

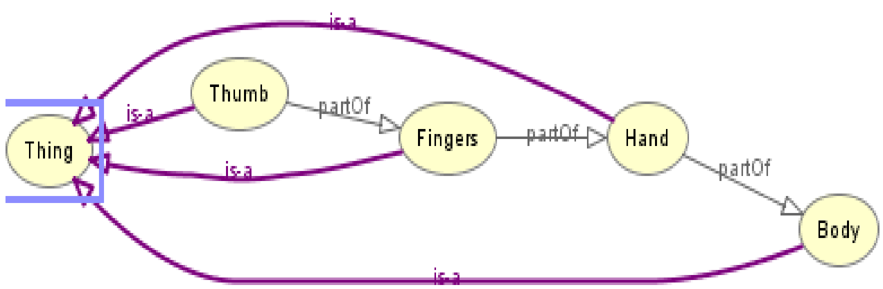

Figure 1 - KG Using "partOf"Figure 1

Figure 1 depicts a Hand class or keyword in a KG that goes beyond standard "is-a" relationship, allowing support for a "partOf" custom relationship so that a better understanding of how a hand relates to fingers and a body.

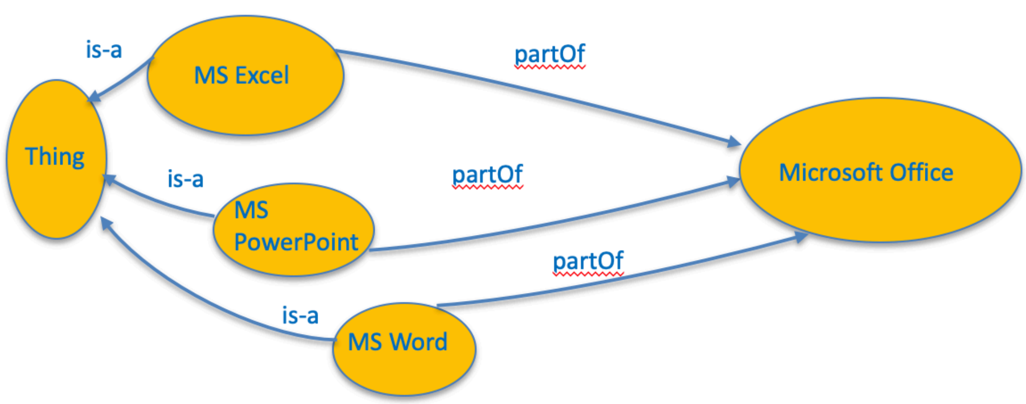

Figure 2 Excel Using "partOf"Figure 2

Similarly, Figure 2, depicts a KG that better shows the relation between Microsoft Office and Excel, PowerPoint or Word components.

Both examples above point out a flaw in existing ATS software as these software packages do not account for matching based on synonyms of keywords. If ATS modified their software to allowing matching based on keyword synonyms, pitfall like those identified in section 1.1.1 would be reduced or eliminated. As we show in section 4, the algorithm adjustment mention could lead to improved matching.

1.4 Expected Contribution

The main objective of this research is to show that improvements are possible when it comes to matching job providers and job seekers. I used a framework that starts with creating a list of keywords from multiple sources related to a specific job (e.g. Cloud Software Developer). The next step involved using keywords as the foundation for constructing KGs. KGs created in this research extended the ontology beyond the “is a” relationship to include “part of “ and other relationships as well as allowed for keyword synonym matching. These adjustments led to improvements in the matching process as outlined in section 4.

The same framework that was used for job and resume matching can be used to find matches in the medical, academic and other industries. I describe similarities in section 4. To that end, this dissertation makes the following contributions:

1. Per section 4, there is evidence that the RA improves the matching process via the used of synonyms of keywords and other filters when searching through resumes
2. The improved matching significantly reduces the amount of time that Human Resources (HR) professionals spend reviewing resumes to decide on a short list of candidates who should be interviewed
3. Given the expanded search on keywords, the short list of candidates should include higher quality candidate based on broader relationships
4. KG created also serve as visual material that can be used to help reduce training time when a HR recruiter takes another job
5. KG also supplements a recruiter's knowledge or compensates a recruiter by providing a visual of deep industry knowledge resulting from relationship that are not always intuitive
6. The framework provides a visual starting point for employees in the medical and academic fields who may have similar assistance in sifting through text to find to a match on keywords and their synonyms

1.5 Dissertation Roadmap

This research is structured as follows: Chapter 2 discusses how pre-processing resumes or getting to a short list of candidates to interview has evolved over the years. Initially, the process was very manual and HR personnel spent hours trying to identify the correct candidates. Automation improved the process, and ATS software led to more improvements. At the end of chapter 2, I point out that KGs which are able to model relationships beyond the classic "as-is" relationship allows further improvements in resume matching. Chapter 3 discusses how KGs can help overcome problems and challenges mention in chapters 1 and 2. In the chapter, a Framework is introduced which describes the process of identifying keywords that help in the creation of KG keyword and keyword synonyms that are used to improve matching between a resume and a job description. HR personnel can also use the KG to refine the short list of candidates produced by the Python Analyzer (PA). The KG also provides benefits of visualizing custom relationships beyond the "as is" relationship found in traditional ontology. I highlight efficient resume pre-processing for a Cloud Software Developer and identify key terms that professional and lay-people use to describe skills and requirements. Chapter 4 discusses results of an experiment that processed approximately 120 resumes against 3 job descriptions related to a Cloud Software Developer. Findings from the experiment confirmed that there is an improvement in match percentages after KG keyword synonyms are applied to our matching algorithms. Chapter 5 discusses conclusions, research contributions, and potential future works.

Chapter 2

Job Interview Pre-processing and Knowledge Representation Methods

HR departments have invested a lot into ensuring that the best and brightest individuals are hired into their companies. “In fact, recruitment has consistently been identified as one of the most impactful HR functions.” [29] The increasing role of technology has led to improvement in identifying candidates for hire. Automating the search through large volumes of resumes has led to improvement over the manual searches for candidate. However, there is still room for improvements via the use of KG and synonym matching when searching through large volumes of resumes. Getting companies to use technology to find candidates for hire has been a step in the right direction, but there is still a substantial number of companies that are not embracing technology for help, or are using technology in limited ways: “…it is a fact that 47% of companies have dated HR software solutions (Bersin by Deloitte), and several others restrict the use of software tools to job advertising and social networks. [5]”

2.1 Job Application Pre-processing methodologies

Job Application Pre-processing (JAP) will be used interchangeably with Application Tracking Systems (ATS) as they both improve a company’s chances of finding the right candidate for a job vacancy through the use of software. JAP involves receiving resumes, then searching through resumes in an attempt to find a short list of candidates whose credentials warrants further evaluation through the interview process. The process of searching through resumes can be manual, automated or combination of both.

2.1.1 The Goal of Job Application Pre-processing

Each company has a process of recruiting and selecting candidates by filtering and screening resumes against job descriptions developed specifically for their companies. Hiring the wrong person can be costly: “A recent study by the Society for HR Management found that the cost of hiring someone for a wrong intermediate position is approximately $20,000 and the cost of hiring the wrong person for a senior manager’s position is $100,000” [28]. Thus, the goal of JAP is to maximize “hits” and avoid “misses” as described in Figure 3 below. Hits are accurate prediction of which resumes best match a job description and misses are inaccurate predictions of matches.

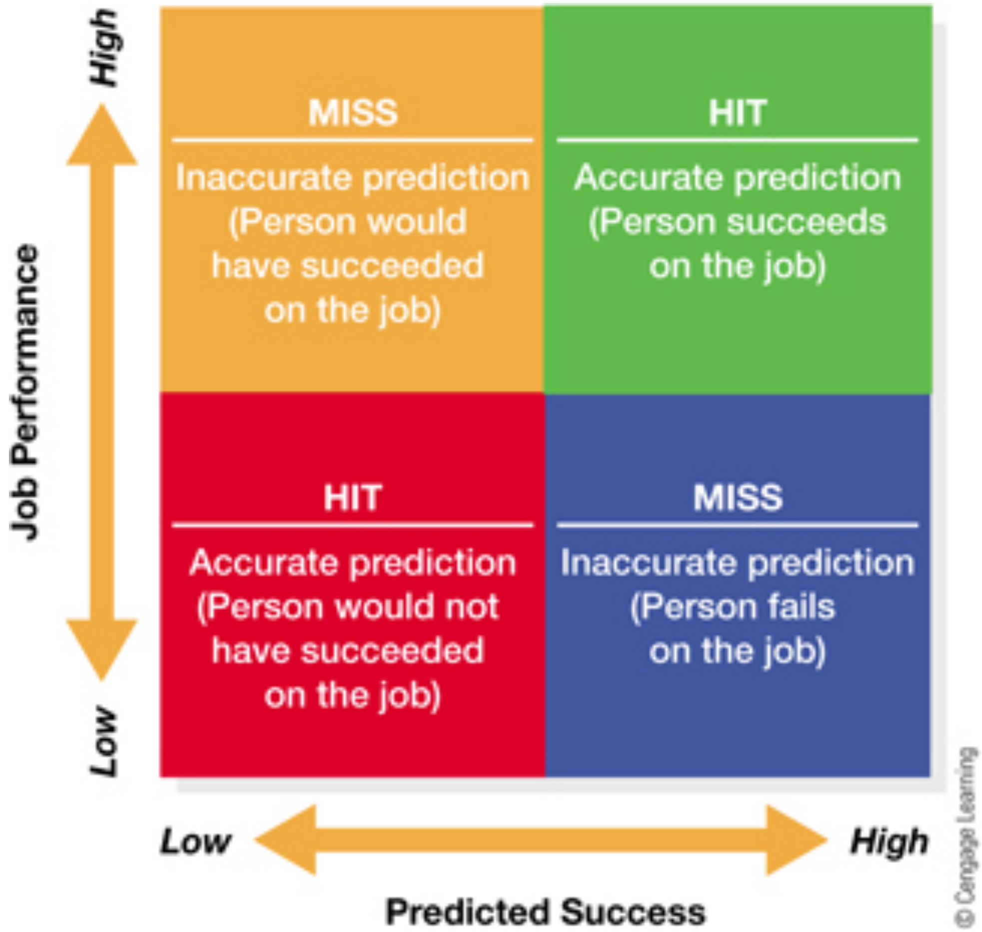

Figure 3 - Job Application Pre-processing (JAP)Figure 3 [28].

Over time, companies are able to identify a set of skills and attributes in their hires that will most likely help a new hire be successful. These skills and attributes ultimately result in a list of keywords associated with education levels and are technologies being used by a specific company. I used some of these keywords to create a visual KG that shows relationship amongst the skills and attributes.

2.1.2 Manual Resume Searches

As mentioned earlier, the traditional JAP involves collecting resumes as input and manually reviewing those resumes to produce a short list of candidates best suited for further consideration through interviews. To that end, the manual nature of collecting resumes, reviewing them and filtering resume down to a short list can be time consuming. For example, consider the HR recruiter who goes to conferences and collect hundreds of resumes for a small number of openings that exist in his/her company. These HR recruiters are instructed by hiring managers to look for certain "buzzword" or keywords when speaking to potential hires or reading through resumes. Because of the manual nature of scanning for keywords, HR recruiters often have deadlines to produce a short list of candidates. The limited amount of time to review candidate may result in selecting candidates who may not be the best fit for the organization's culture. "Traditional hiring processes force recruiters to choose between speed or quality of hire, specifically if you have a large candidate pipeline." [8] To do an objective quality job of assessing candidates, recruiters have to spend time obtaining a full view of a candidate (i.e., does personality fit culture?) which leaves less time for other administrative work. Research shows that "The old way resulted in too many commitments and not enough time or resources to achieve consistent results in any one area." [32]

For organizations that receive a large number of resumes, recruiters use resume skimming skills to search through a large number of resumes in a short amount of time. "Studies have shown that the average recruiter scans a resume for <u>six seconds</u> before deciding if the applicant is a good fit for the role." [2] Skimming resumes allows recruiters to search through lot of resumes, but candidates selected for the short list may be of poor quality or lack some desired skills. Despite fierce competition for talent, the six second "Eye-Tracking Study" referenced above pointed out that instead of focusing on key job requirements, recruiter focused on superficial details like resume layout, job titles, text flow and keywords. This is problematic because most hiring involves bring people in from outside of the company: “A recent SHRM found that 25% of organizations filled positions with current internal employees and 75% with external hires.” [27]

In addition to being pressed to make quick decisions about who to interview, many HR personnel also lacked domain level expertise or a technical background sufficiently deep enough to evaluation a candidate’s technical background. Many HR personnel struggle with understanding technology beyond that of the user level when it comes to using social media (FaceBook, LinkedIn, etc.) or the cloud for recruiting. “In today’s world, technological skills are no longer a bonus, but a necessity for every HR professional. New technologies are coming out on an almost daily basis which means HR needs to be more agile now than ever before.” [3] [4]. The lack of deep technical expertise can present a problem for recruiters who are tasked with recruiting technical professionals.

2.1.3 Automated Resume Searches

Automated JAP or ATS involves collecting resumes as input with the use of software to produce a short list of candidates best suited for further consideration through interviews. ATS software has emerged as the standard for streamlining recruitment and for being the source for creating the short list of candidates recommended for interviews. Today, approximately 95 percent of Fortune 500 companies rely on ATS software, and "75 percent of job applications are rejected *before* they are seen by human eyes." [2]

For organizations that receive a large number of resumes (i.e., thousands per week), ATS software acts a gate keeper scanning resumes for keywords of interest so that resumes associated with candidates who are least qualified for the job are eliminated from the applicant pool. Since software is doing the work, it is more efficient than humans and can produce results faster. Figure 4 below shows that ATS software has been a big help to organizations that receive an average of 250 resumes for a job that will invite approximate 4 or 6 candidates in for an interview.

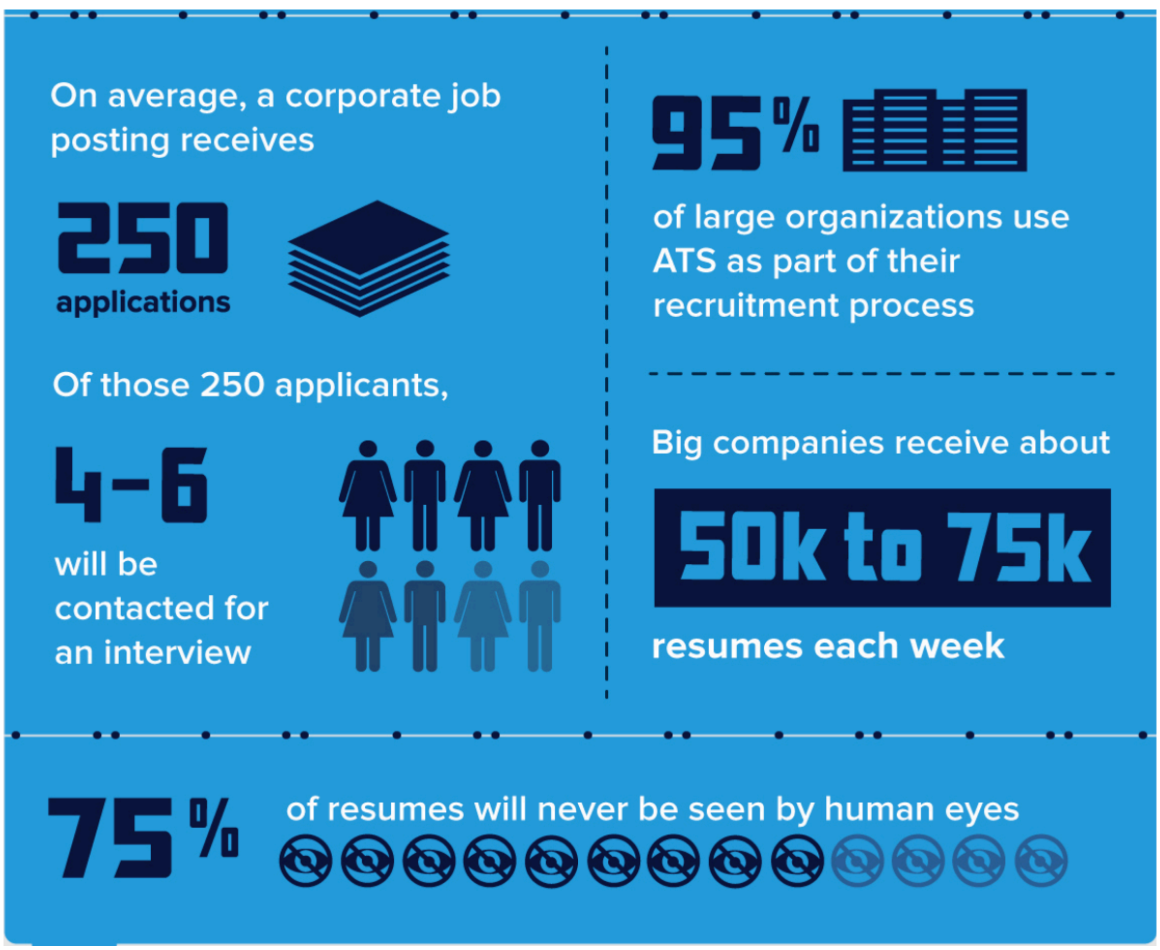

Figure 4 Time Spent Reviewing ResumesFigure 4 [2]

A July 2018 article by Skeeled indicates: "We know for a fact that each job opening attracts 250 resumes (Glassdoor) and that a great number of these resumes are unqualified and junky (MRINetwork 2017 Recruiter Sentiment Survey). Moreover, consider time-to-hire the top measurement of success." [31]

Automation is a time saver for employers that find it a waste of time to manually review resumes of unqualified candidates. In 2017, approximately 400 MRINetworks offices were used to survey employers regarding a number of topics including barriers to identifying qualified talent. Figure 5 below reflects some results from the survey.

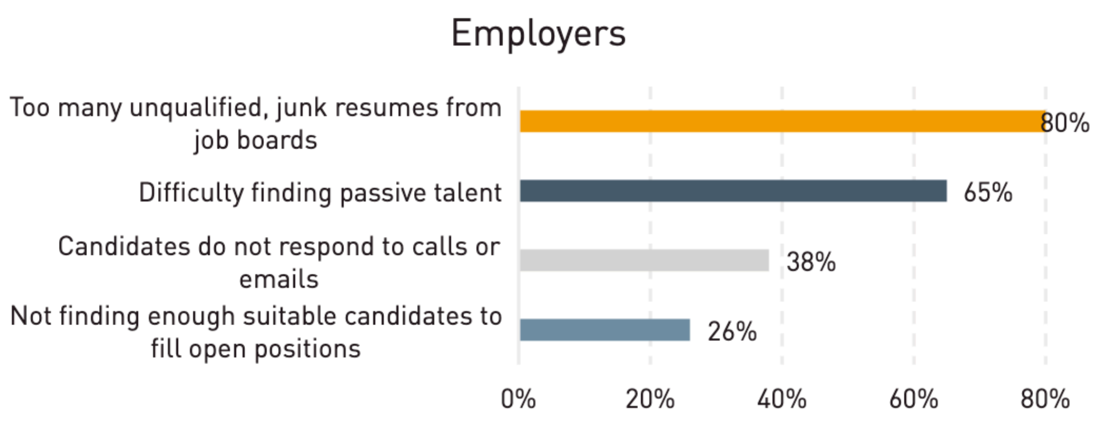

Figure 5 What Barriers are There to Identifying Qualified Talent?Figure 5 [30]

One study shows that automation provides benefit to both employers and job seekers: "Job seekers would benefit from the increased market transparency. Employers would benefit from reaching more potential applicants, saving money on publication fees and from cutting costs by automating the preselection of applications." [6]

2.1.4 Advantages to shifting from manual matching to ATS matching

The trend towards streamlining and automating the process of matching content in job description with content in resumes has led to improvements when compared to the manual process. Organizations experiences business growth, insufficient recruiting staff/team, and high turnover have highlight the following reasons for moving from the manual matching to automation or ATS matching: [22]

1. **Save time** – Firms no longer spend time reviewing resumes of unqualified candidates which could be up to 75% of applicants. Manual input of data into spreadsheets, and other tools is also eliminated or reduced
2. **Be more efficient** – ATS allows for standardization of candidate data making it easy to track applicant, share applicant information and search for information based on location, skill level, educational background, etc.
3. **Be more productive** – ATS allow all involved in the recruiting process to share feedback about candidates using standard templates which helps in decision making
4. **Improve quality of hire** – Since the ATS eliminate unqualified candidates from consideration, the quality of those that eventually get hired should improve. This should also help with retention.
5. **Use Performance Reports** – ATS data provides metrics like which recruiting forum (i.e., referrals versus career fairs, etc.) has the highest return when you consider candidates found versus cost associated with the event (i.e. yield ratios, etc.)

6. **Provide better Candidate Experience and boost your Employer Brand** – An ATS can be configured to automatically send out emails to candidates not selected to move on, or not selected to move to the next hiring phase. This allows applicants to be notified of where they are in the hiring process and helps with the company brand.
7. **Competitor positioning** – In order to remain competitive with peers, companies need to leverage ATS technology the same way that peer companies are doing.

Figure 6 below provides a list of the top ATS providers.

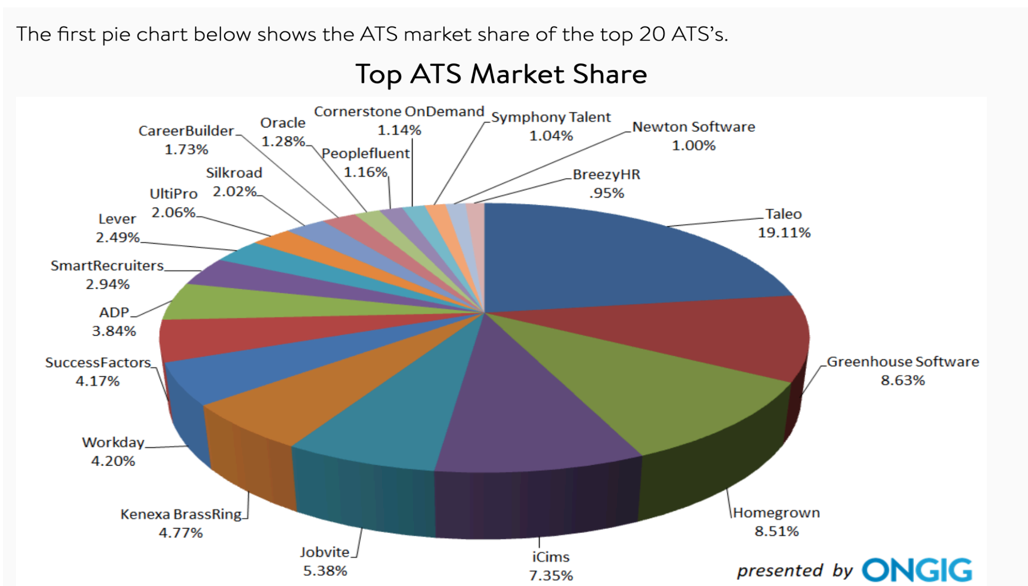

Figure 6 ATS Software ProvidersFigure 6

Note: Companies like Google, Facebook, Apple, Starwood and Wyndham are included the "Homegrown" category because at some point they have use a homegrown-built ATS at some point.

I should point out that “Today, approximately 95 percent of Fortune 500 companies rely on ATS software to help streamline their recruitment process.” [18] ATSs were initially designed to assist HR departments in large corporations. However, the expanding list of features have ATS packages very popular, and many companies see adopting and ATS as a way to stay competitive. In addition to resume matching, some ATS software can assist with candidate credit checks, code reviews and other features that can improve the overall hiring process as well as improve a company’s brand name.

2.1.5 Challenges associated with ATS matching

Although the previous section highlighted advantages associated with automation and ATS software, there are drawbacks we should also consider.

A drawback worth noting is that ATS software relies heavily on keyword matching which does not make allowances for synonyms and other important relationships (i.e. like the “part of” relationship that will be discussed in other parts of this paper). I show in the experiment section of this paper that there is an improvement in match rates when the RA ATS is adjusted to account for synonyms and other custom filters. “Reasoning with keywords, and in particular ascertaining what keywords are about, are important issues in computing” [17]

Another drawback is expanding features is increased complexity in integrating ATS software with third-party software of clients. For example, if an ATS systems allows the use of Google calendaring for a candidate to schedule an interview, the interface between

Google Calendaring and the client's ATS will require testing that may or may not work seamlessly.

The cost to switch from one ATS provider to another could also be a factor. The more an ATS is integrated with a client's computer systems, the more impact there could be if a company decides to switch vendors. Code written to exploit interfaces could be costly. If the integration is not done properly, there could also be an impact to the user experience eroding some of the goodwill brought on by implementing an ATS in the first place.

Finally, although ATS are used, many non-technical people are not aware of them. For those who are aware of ATS, they realize that the logic underlying ATS vary by vendor making it practically impossible to have a resume that is optimized for all ATSs. Each company also uses different words to describe jobs making it even more difficult to optimize for all jobs. Thus, optimizing a resume based on keywords may not provide the advantage initially expected.

2.2 Forms of Knowledge Representation Methods

2.2.1 Ontology-Based

Ontologies provide a visual representation of knowledge. Think of ontologies being the practical implementation of the expression "a picture is worth a thousands words." Ontologies are used to help with knowledge sharing by creating a visual representation of concepts that aid people in arriving at a common understanding of an area or domain. The clarity that ontologies provide help to communicate difficult concepts. Ontologies provide for a common vocabulary and visual that allows people to achieve a shared understanding of terms, means and relationships in a particular area or doman of knowledge. “The ontology defines concepts, relationships, and interlinking of concepts using relationships.” [1]

Key components of ontology are that they:

1. Capture concepts in visual formats making the concept easy to understand
2. Hightlight relationships (i.e. parent versus child, etc.) in diagrams
3. Allow knowledge sharing and reuse which improves communications
4. Describe assumptions (i.e. Mary is Anne’s daughter, therefore Anne’s son is assumed to be Mary’s grandson)

Ontology are usually viewed as being hierarchical. For example, a class like Computer Programmer can be thought of as a class in the software doman and a Python Programmer would be a subclass or more specific type of programmer.

The five components of an ontology are classes, relations, functions, formal axioms and instances: “

a) Classes represent concepts, which can be considered generic entities.
b) Relations represent a type of association between concepts of the domain.
c) Functions are special case of relations.
d) Formal axioms serve to model sentences that are always true. They are normally coherent description between Concepts / Properties / Relationships is logical expressions.

e) Instances are used to represent elements or individuals in an ontology." [26]:

Ontologies can be created in multiple languages including Web Ontology Language (OWL) which is recommended by the World Wide Web Consortium (W3C). Below are examples of the three key areas in OWL.

Individuals: are instance of a class. For example, Madison is an individual or instance of a Student class or Italy is an individual or instance of a Country class. As well as Figure 7 shows the representation of Individuals.

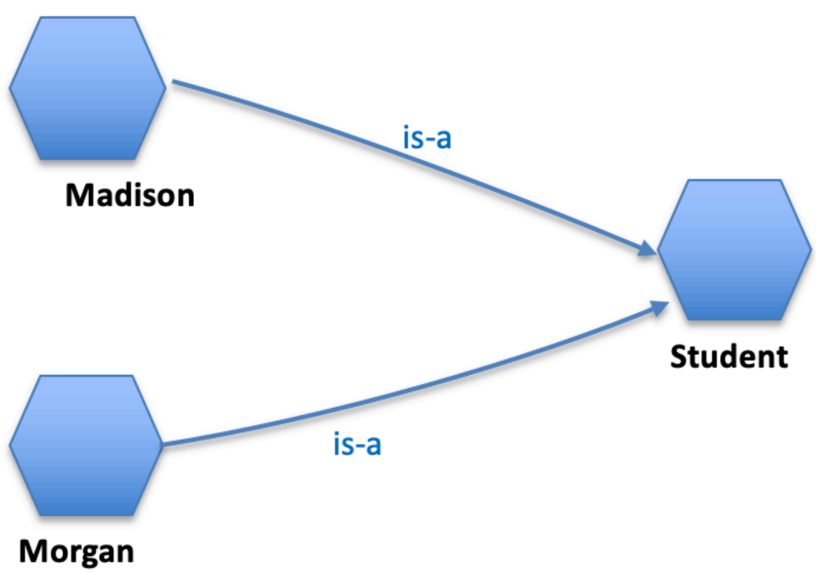

Figure 7 - Representation of IndividualsFigure 7 [26]

Properties: relate two individuals together. For example, individual Madison has a sibling who is Morgan – see Figure 8 which shows the representation of Properties.

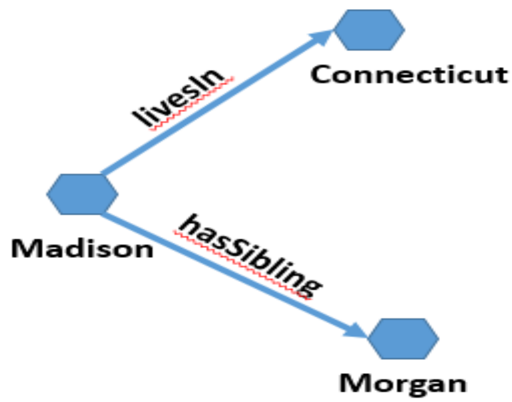

Figure 8 Representation of PropertiesFigure 8

“Classes: can be described as a set that has individuals as members. Examples of a class are Person and Pet. Individual Madison in class person has a pet named Happy in class pet – see Figure 9 which shows the representation of Classes.

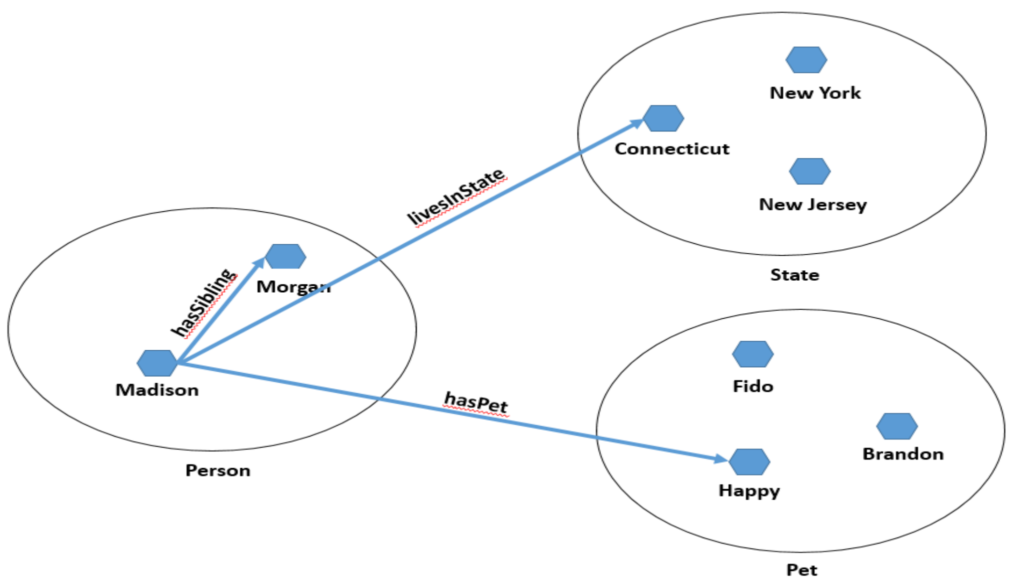

Figure 9 Representation of ClassesFigure 9

OWL uses the Triple to describe the relations between two entities. Figure 10 depicts this relationship.” [26]

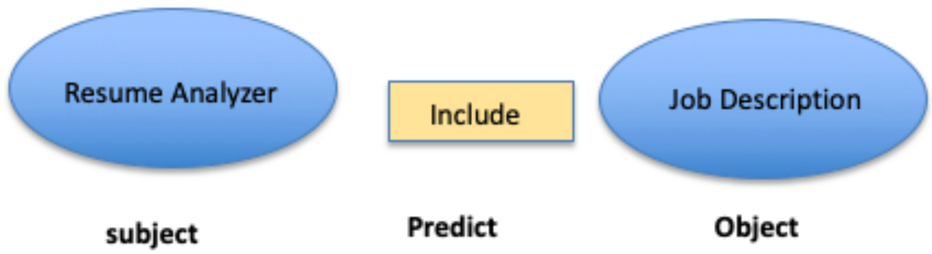

Figure 10 Example of Triple RepresentationFigure 10

Protégé is considered the most popular tool for creating OWL files. There are many versions of Protégé such as Stanford University's version of Protégé. Protégé is limited in it use of only the "is-a" relationship. For example, `"Dog is-a Pet"'. In this triple, "Dog" is the subject, "is-a" is the predicate, and "Pet" is the object. With "is-a" being the only relationship between two classes, properties may have to be used to capture a more vivid knowledge of the domain of interest. Figure 11 depicts an ontology created with the "is-a" relation of a class "Pet" and its subclasses. [26]

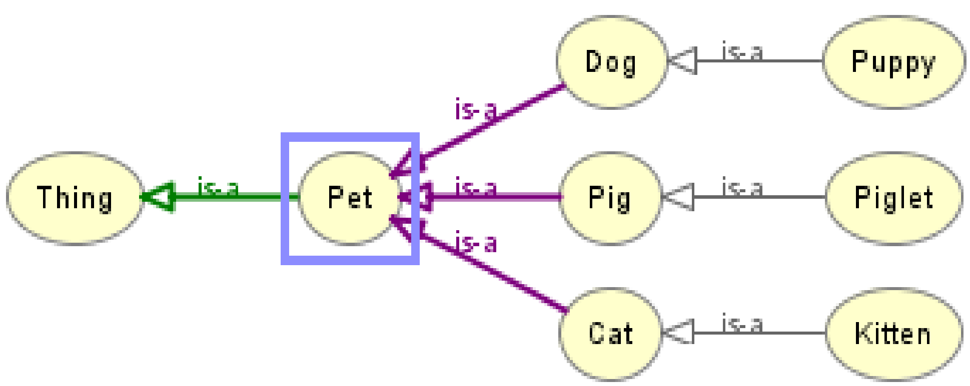

Figure 11 Pet Relations Using "is-a" PredicateFigure 11 [26]

Although tools like Protégé allow for the creation of visual images, or ontologies, restrictions associated with the tool does not allow for the creation of KG with richer relationships among entities. Plus, there is a considerable amount of manual work when it comes to customizing ontologies: "… ontology customization and integration tasks still require a considerable amount of manual work, even when using common representation languages like XML schemes or OWL." [20]

2.2.2 Knowledge Graphs

As we have seen in section 2.2.1, although Protégé and OWL are regarded as the industry standard in knowledge representation, the "is-a" only relationship severely limits the creation of ontologies with custom relationship. Custom relations vary by domain, and the partOf relation is important in supporting concepts related to knowledge representation. This research uses Pace University's Knowledge Graphs (KG) to support a set of custom relations unique to matching keywords found in resumes and job description. Job descriptions and resumes from job seekers use keywords that have synonyms which are encoded in the KG to increase the chances of a match on keywords.

Pace University's KG allows for custom relations that makes OWL more robust. The custom relations benefit OWL through the use of minimal syntx extention that allow OWL to continue interfacing with existing tools.

Figure 12 below depicts a Hand class with custom created relations. It is evident that in addition to the “is-a” relation, the “partOf” custom relation was used to relate two classes in a meaningful way.

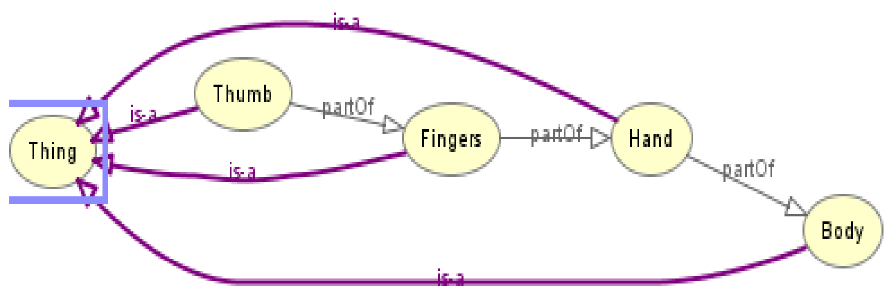

Figure 12 Hand Relations Using “partOf” RelationFigure 12

A word of caution regarding KGs

There are many search routines that attempts to aid in the matching of job seekers with job providers. Some routines have used knowledge graphs in an attempt to improve results [24] . For example, some queries tried to use relevance data (i.e., scores, etc.) in conjunction with knowledge graphs to improve search results. Using a visual aid like a knowledge graph is good in that it helps in identifying relationships between relevant entities. Although knowledge graphs (KG) are extensively used [34] and are good at visually representing concepts, there are drawbacks associated with KG’s expression of *Resource Description Framework* (RDF) language. It is difficult to express visual rule with RDFs and positioning the RDF is arbitrary.

When using KG for navigation, nodes are considered concepts and edges represents relationships. Given this, ontologies can be visualized as graphs and can play the role of conceptual structure for navigation [10] . A node in our KG could be a job description that has relationships with a concept like education requirements. The primary benefit of the KG is that it allows a visual to help understand reltaionships

Summary:

This chapter references experts who point out that HR departments see bringing in new talent as a critical activity. Experts also point out that the volume of resumes received by companies have become so large that automated matching and filtering of resumes are needed. ATS packages provide automated matching and filtering, but ATS' reliance on keyword leaves room for improvment during the keyword matching process. To address problems associated with keyword matching, visual aids like Ontology digrams and KG diagrams can be used. It was then explained that Ontology diagrams are limited to modeling based on the "is a" relationship where KGs extends ontology diagram to include rich relationships like "part of" and other custom relationships.

Chapter 3 Efficient Resume Pre-Processing Framework Inspired By Knowledge Representation

3.1 Python Analyzer Framework Overview

At a high level, Figure 13 below provides an overview of the Python Analyzer (PA) Framework.

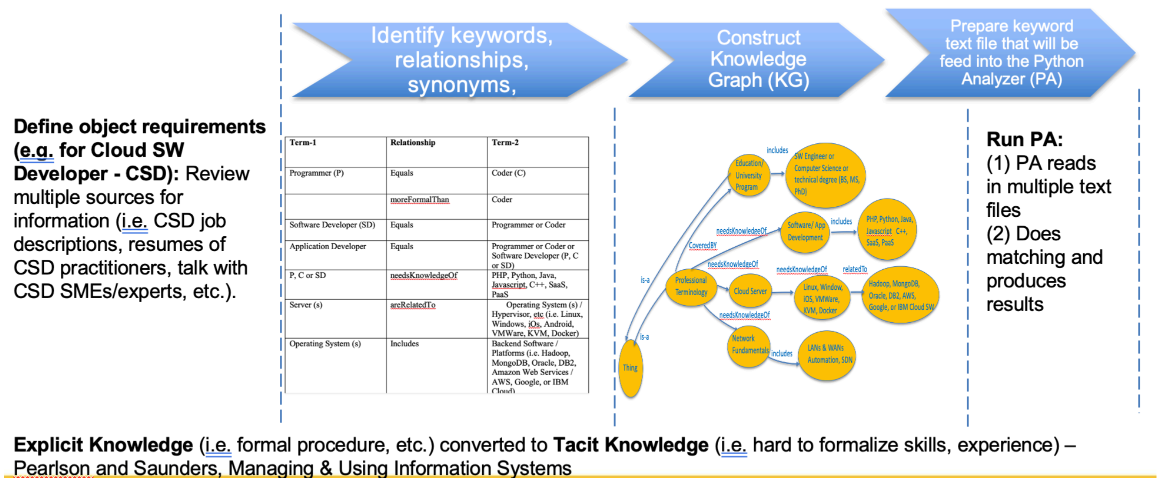

Figure 13 Python Analyzer (PA) FrameworkFigure 13

On the far left of Figure 13 is a column that captures requirements for the object that may be of interest to a user. For this research, we are focused on requirements for a Cloud Software Developer (CSD). To that end, CSD information from multiple sources (i.e. cloud job descriptions, resumes of cloud practitioners, etc.) is collected and reviewed. In analyzing collected information we identify relevant keywords and construct a table like the one listed below showing important relationship like "CSD equals Programmer or Coder." The table is used to create a Knowledge Graph (KG) which captures custom relationships. The far-right column of Figure 13 shows that a superset of keywords are captured in a text file and feed into the Python Analyzer (PA) for comparison against CSD job descriptions from IBM, Amazon or Google. The output from each job description comparison results in a common set of keywords that are shared between the text file and job description. The common set of keywords are then compared against resumes from each candidate applying for the CSD job. This kind of knowledge conversion is referred to as Internalization as it supports "Explicit Knowledge being converted to Tacit Knowledge," [21]. Pearlson and Saunders define Explicit Knowledge as knowledge that comes from organized procedures, books, reports or articles whereas Tacit Knowledge comes from experiences, beliefs and skills that are hard to formalize [23]. The KG represents know-how or previous learnings that is captured in a graph and ensure that filtering of resumes is based on desired custom relationships.

After the PA has determined match percentages between resumes and a job description, it prints out a shortlist of resumes with the top 10 match percentages. An HR Recruiter can use the short list of resumes to select the top 3 to 5 resumes that he/she would like to continue through the hiring process. This activity can be thought of as "Explicit Knowledge being converted to Explicit Knowledge" [21] as knowledge is combined and may produce new knowledge.

Subsections that follow will address the Python Analyzer algorithm as well shows how KGs used in the framework have been decomposed into 3 parts from high to lower levels of knowledge representation. This section also highlights examples of how keywords and the KG are used to avoid common problems for job description for Amazon, Google and IBM. Finally, examples are given of how the PA framework can be used to improve matching in medical, academic and other industries.

3.1.1 Python Analyzer Algorithm and Implementation

The Python Analyzer (PA) algorithm is displayed in Figure 14 below. When the PA is initiated, the person running the program is prompted to enter a “1” to execute instructions for IBM, a “2” to execute instructions for Amazon or a “3” to execute instructions for Google. Table 4 has pseudo code which describes the PA algorithm further.

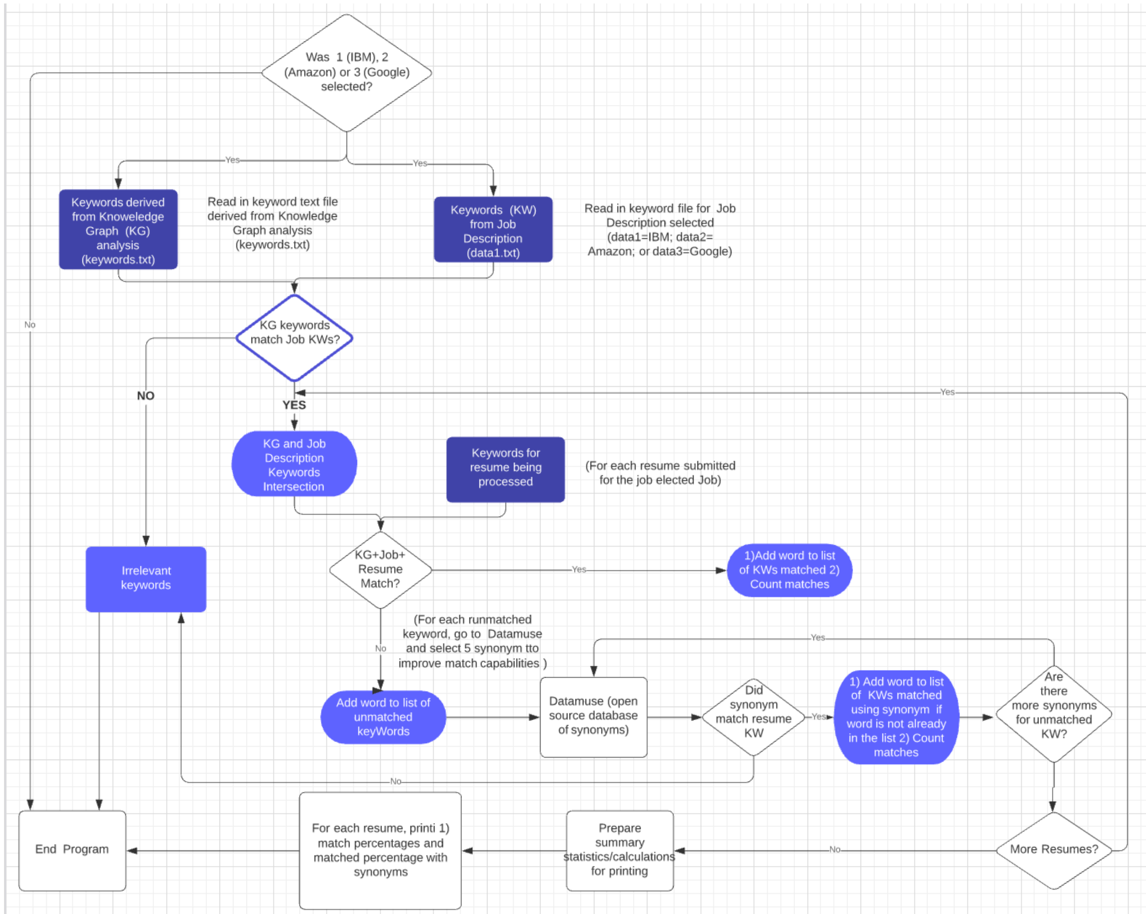

Figure 14 Python Analyzer Algorithm

Table 4 below has pseudo code which describes the PA algorithm in more detail.

Table 4 Python Analyzer Pseudo Code

1: **Prompt user to enter a “1” to execute instructions for IBM, a “2” to execute instructions for Amazon or a “3” to execute instructions for Google**

2: **Read in job description based on user selection**

3: **Read in keyword file based on Knowledge Graph (KG) analysis**

4: **Create list of keywords based on keywords (kw) from the KG analysis file that matches words in the job description**

5: **For as long as there are resume, read each resume and compare each word from the list of keywords created in step 4 to each word in the resume**

6: **if there is a keyword match**

7: **Add** the keyword to the list of matched keywords; store for printing later

8: **Count** the number of matched keywords; store for printing later

9: **else**

10: **Add** the unmatched keyword from the list in step 4 to a list of unmatched keywords; store for printing later

11: **for** each unmatched keyword, go to an opened source database titled Datamuse and find 5 synonyms for the unmatched keyword

12: **for** each synonym

13: **if there is a match to a word in the resume**

14: **Add** the keyword to the list of matched keywords using synonyms if the keyword is not already there; store for printing later

15: **Count** the number of matched keywords using synonyms

```
if the keyword is not already there, store for printing later
                end if – synonym match processing
            end for – no more synonyms
        end for – no more unmatched keywords
    end if – keyword match processing
end for – no more resumes
Prepared summary statistics/calculations for printing
Display or print before and after match percentages
Display or print top 10 best matches
```

As you might suspect, the Python Analyzer was implemented using Python code. Appendix D has source code for the Python Analyzer.

In the remainder of this section, I will describe how Knowledge Graphs enhances existing ontology to result in improved matching between resumes and job descriptions. I will start by reviewing steps involved in constructing a KG from keywords. The section also shows how KG used in the framework have been decomposed into 3 parts from high to lower levels of knowledge representation. This section also highlights examples of how keywords and the KG are used to avoid common problems for job description for Amazon, Google and IBM. Finally, examples are given of how the PA framework can be used to improve matching in medical, academic and other industries.

3.1.2 Identification of Job Related terms for a Cloud Software Developer

The first step in the PA Framework is to identify keywords for a job that is trying to be filled. In my example, I am using a job description for a Cloud Software Developer. Table 4 below shows relationships between keywords. I should note that only the “is a” relationship can be modeled by ontology.

KG listed in other parts of this chapter were constructed using keywords and relationships highlighted in the Table 5 below.

Table 5 Job Related Terms/KeywordsTable 5

Term-1	**Relationship**	**Term-2**
Programmer (P)	Equals	Coder (C)
	moreFormalThan	Coder
Software Developer (SD)	Equals	Programmer or Coder
Application Developer	Equals	Programmer or Coder or Software Developer (P, C or SD)
P, C or SD	needsKnowledgeOf	PHP, Python, Java, Javascript, C++, SaaS, PaaS
Server (s)	areRelatedTo	Operating System (s) / Hypervisor, etc (i.e. Linux, Windows, iOs, Android, VMWare, KVM, Docker)
Operating System (s)	Includes	Backend Software / Platforms (i.e. Hadoop, MongoDB, Oracle, DB2, Amazon Web Services / AWS, Google, or IBM Cloud)

Network (s)	Includes	LANs, WANs, Automation, SDN
Education	Includes	Training or certification as a 1) P, C or SD 2) in Computer Science or a technical degree .
Cloud – Platform	Includes	Elastic Compute Cloud (EC2) – from Amazon Google Cloud Platform (GCP) Bluemix, IBM Cloud
Cloud – Types	Includes	Public, private, hybrid
Cloud – Features	Includes	Scaling, Virtualization, Load Balancing, Artificial Intelligence (AI), Natural Language Processing (NLP), Scalable

3.1.3 Knowledge Representation via Three Levels of KG

Visuals for PA relationships

Figure 15 and Figure 16 provides a partial visual of concepts among key classes in Knowledge Graphs that will be displayed in this section.

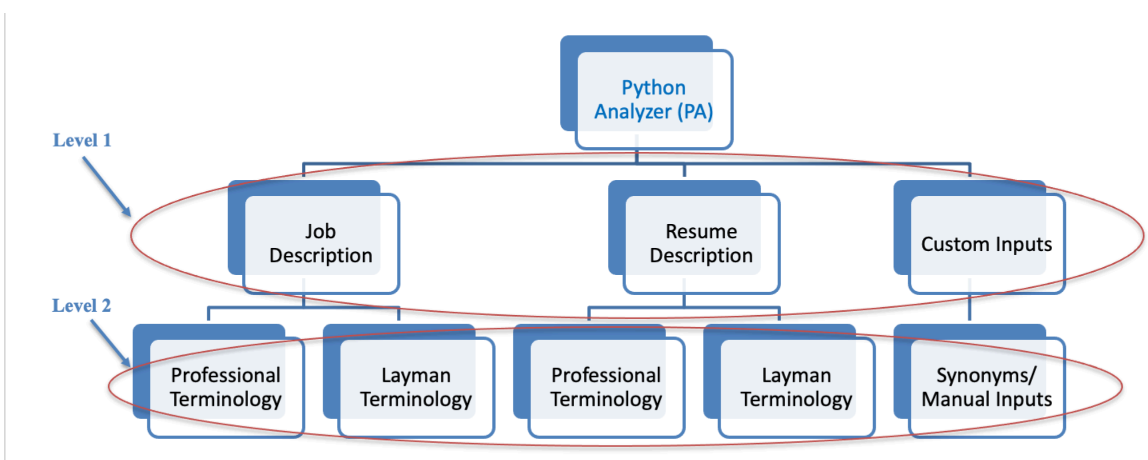

Figure 15 (Level 1 – top part)Figure 15

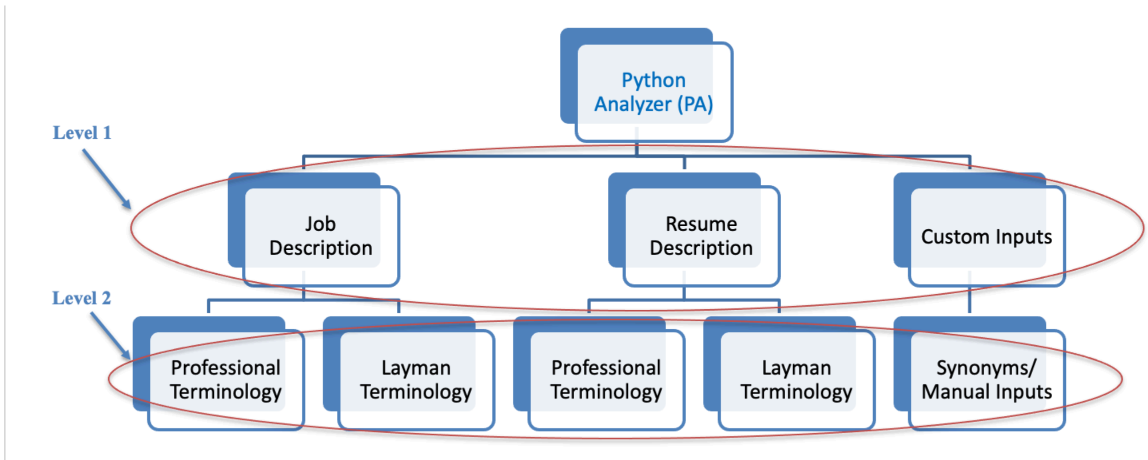

Figure 16 (Level 2 – bottom part)Figure 16

Figure 17 below goes further and includes a KG with relationship among concepts/ classes.

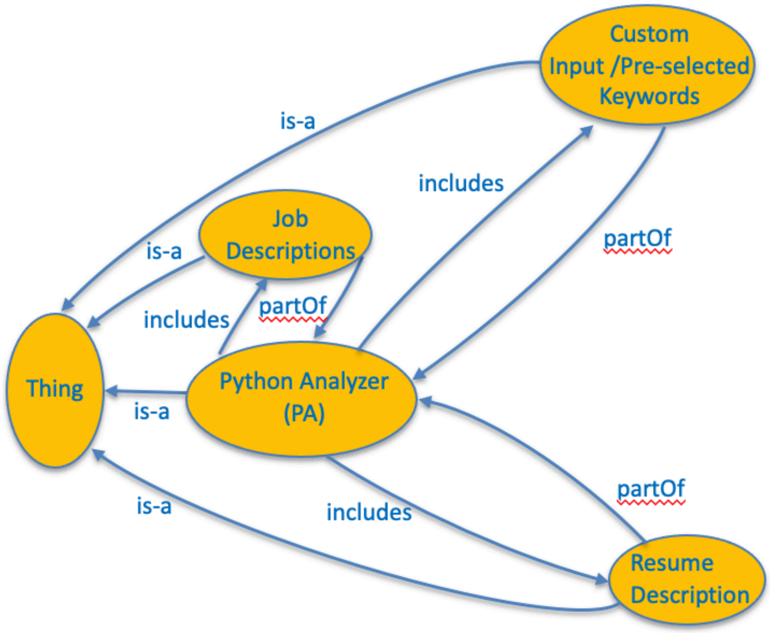

Figure 17 First Level KG (A) – custom relationshipsFigure 17

In Figure 17 the PA is a thing that includes other things like a Job Description, a Resume Description and Custom Inputs. Each thing (i.e., class) is also part of the PA.

<u>Second Level KG Visuals for Python Analyzer (PA) Relationships</u>

A KG of Level 2 in the concept hierarchy is shown below in Figures 18, 19 and 20. Job Description, Resume Descriptions and Custom Input are part to the PA. These areas are further decomposed at level 3.

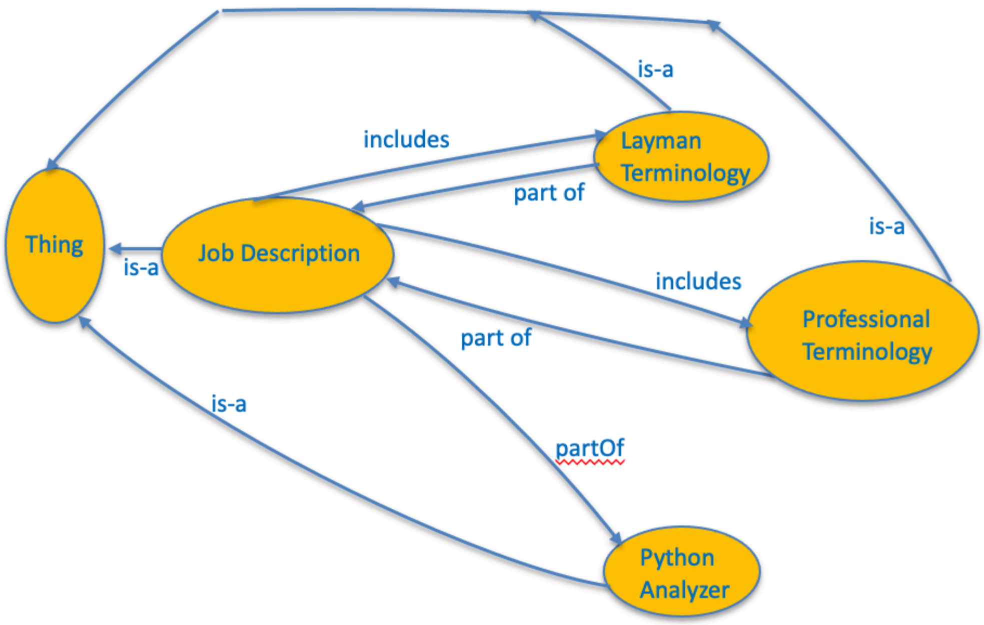

Figure 18 Second Level KG (A)Figure 18

Job Description is a thing and is included in the PA. The KG also includes other things like Layman's terminology and professional terminology.

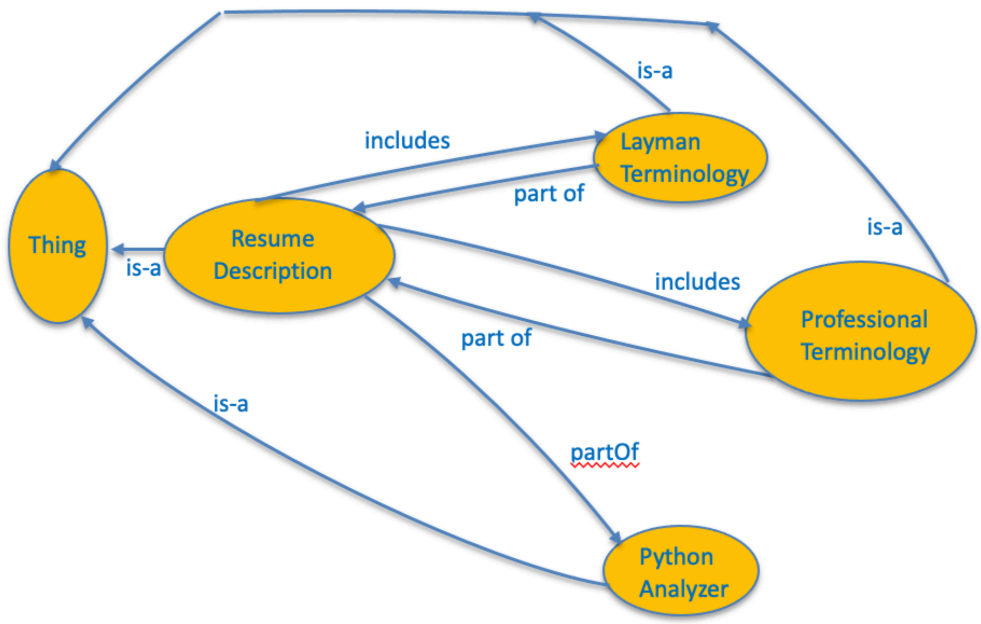

Figure 19 Second Level KG (B)Figure 19

Resume Description is a thing and is included in the PA. The KG above also includes other things like Layman's terminology and professional terminology.

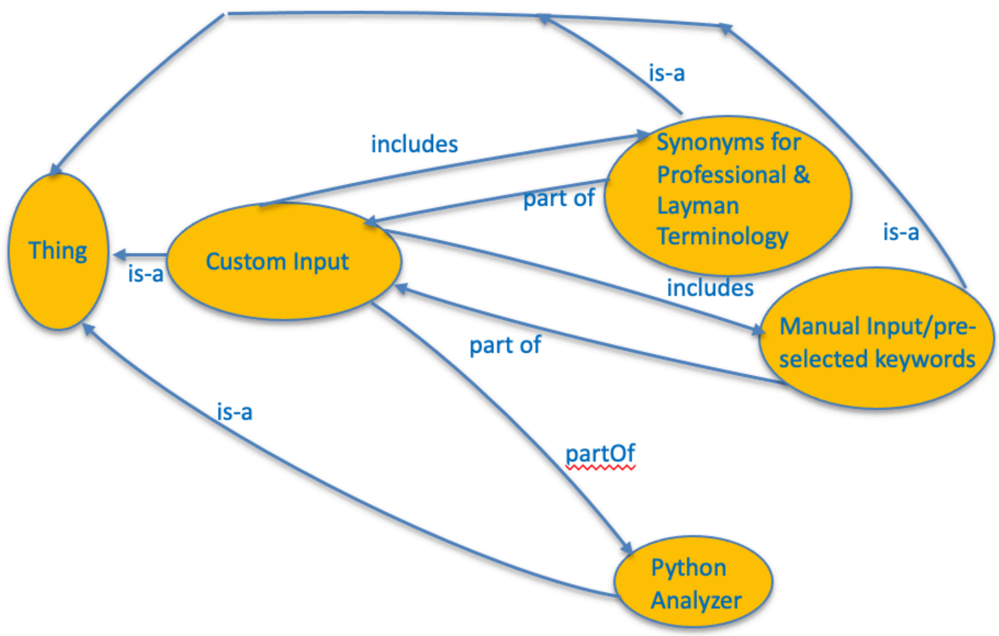

Figure 20 Second Level KG (C)Figure 20

Custom Input is a thing and is included in the PA. The KG above also includes other things like Synonyms for Layman's terminology and professional terminology.

<u>Third Level KG of Python Analyzer (PA)</u>

Figures 20, 21 and 22 provide a partial visual of concepts among classes at level 3 of the visual hierarchy.

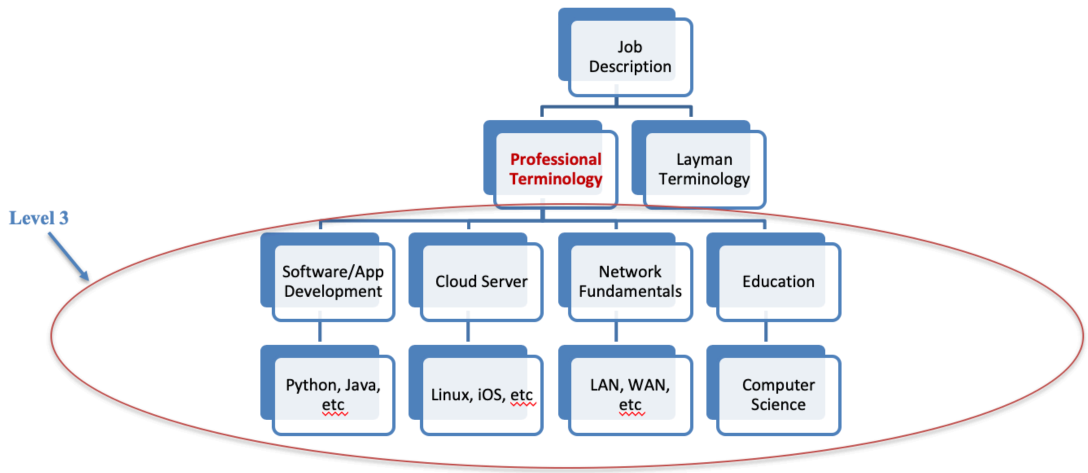

Figure 21 Visual of Concepts for Levels 3 – Prof TerminologyFigure 21

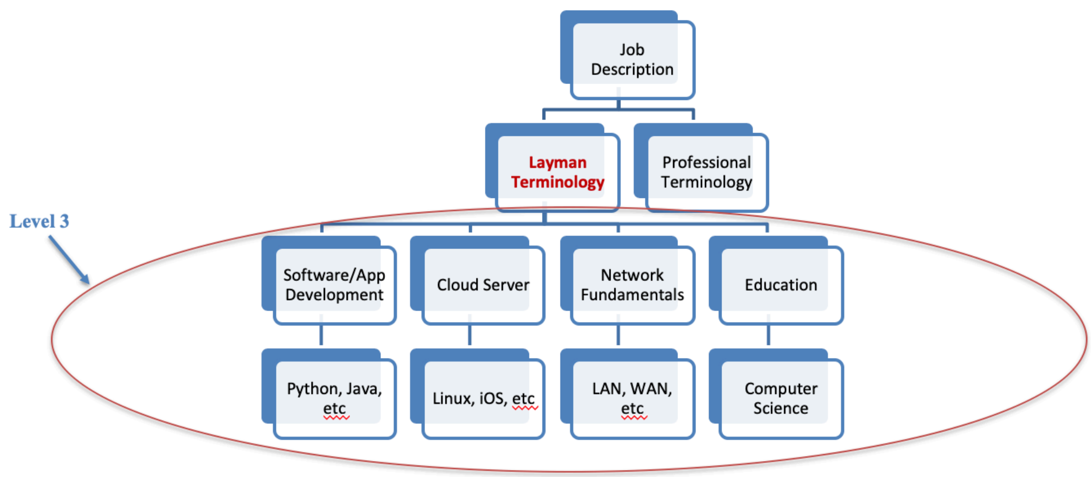

Figure 22 Visual of Concepts for Levels 3 – LayFigure 22

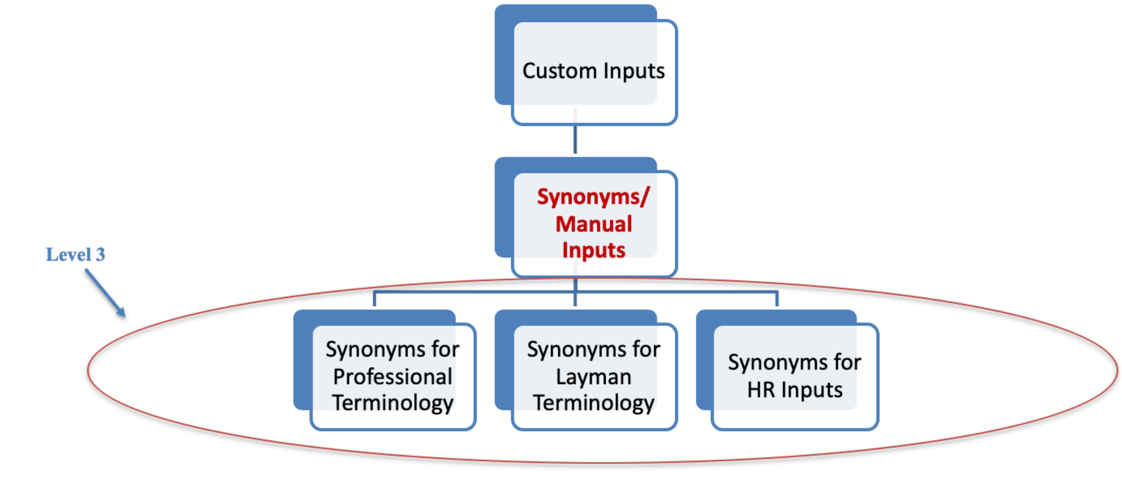

Figure 23 Visual of Concepts for Levels 3 – Custom InputFigure 23

The set of KG that follow provide more detail of relationship among concepts/ classes in the last 3 visuals. Relationships highlighted in KG are used to identify keywords from terminology. Some terms are feed into the PA and used for matching between a resume and a job description.

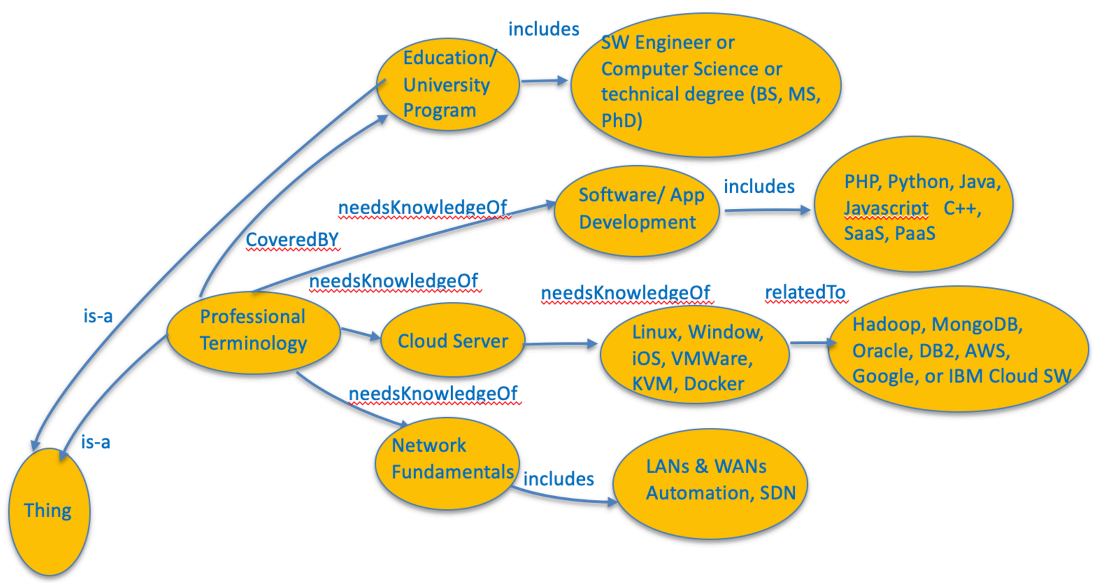

Figure 24 Third Level KG (A)Figure 24

Professional Terminology is a thing and is included in both Resume Description and Job Descriptions. The KG above also includes other things like professional terms to describe Education (i.e., Bachelors, Masters, Computer Science, etc.) Software Development (i.e., Computer Programmer, etc.) Cloud Server (i.e., Linux, etc.), Network Fundamental (i.e., LANs, etc.). Some terms are feed into the PA and used for matching between a resume and a job description.

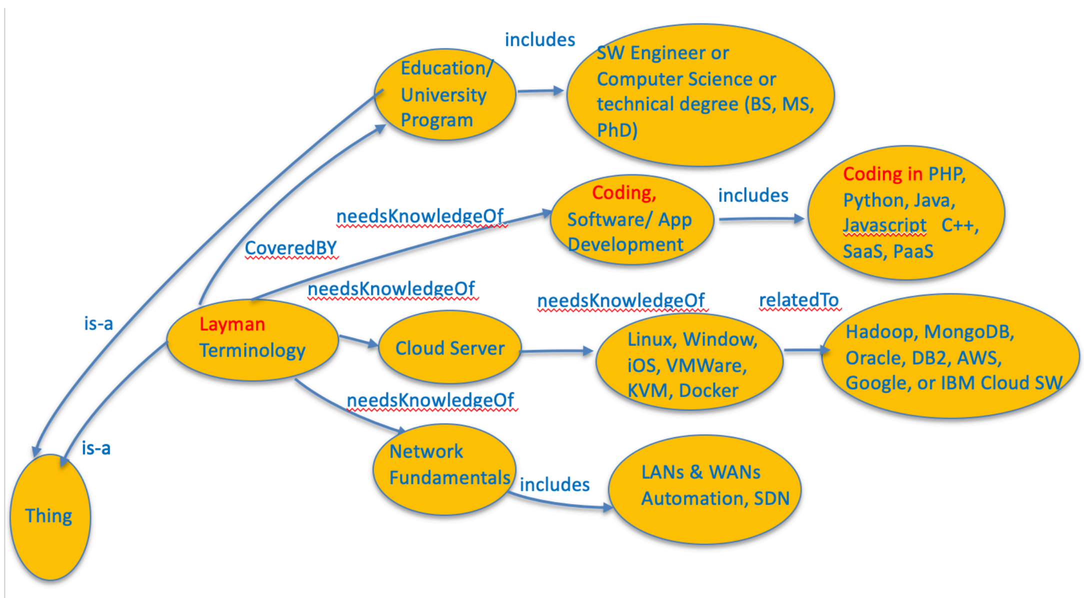

Figure 25 Third Level KG (B)Figure 25

Layman Terminology is a thing and is included in both Resume Description and Job Descriptions. The KG above also includes other things like Lay terms to describe Education (i.e., BA, MS, CS degree, etc.) Software Development (i.e., Coder, etc.) Cloud Server (i.e., Open Source box, etc.), Network Fundamental (i.e., the wire, the net, etc.).

Some terms are feed into the PA and used for matching between a resume and a job description.

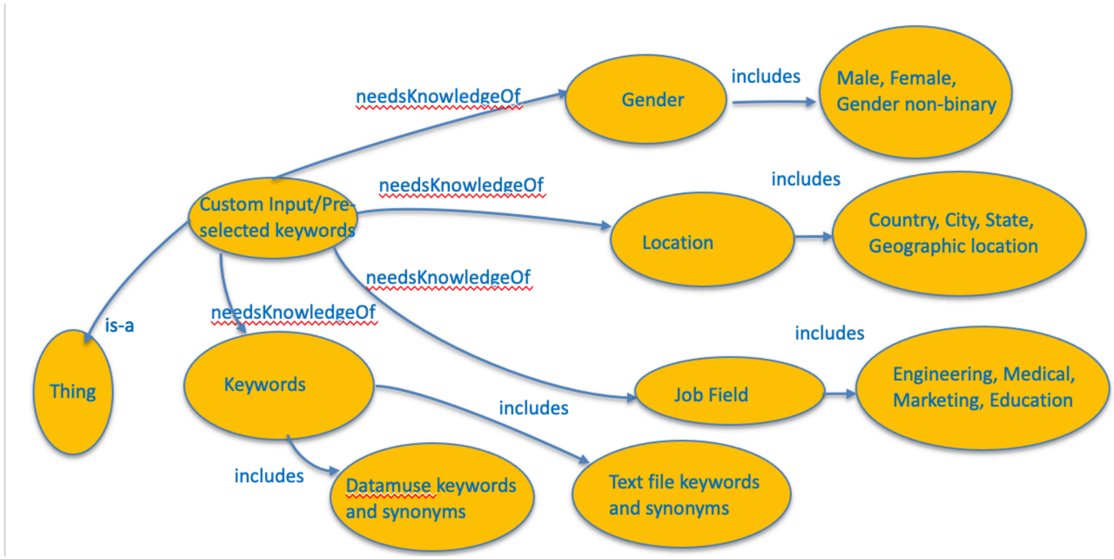

Figure 26 Third Level KG (C)Figure 26

Custom Input/Pre-Selected Keywords is a thing and is included in the PA. The KG above also includes other things like Keywords, Gender, Location, Job Fields. Keywords from the KG text file as well as synonym keywords from the external database (e.g. Datamuse) may also be included in Custom Input/Pre-Selected Keywords.

3.1.4 Using the Framework on Job Descriptions for Amazon, Google and IBM

From the prior section, relationships and keywords in Table 4 lay the foundation for finding matches based on keywords, plural forms of keywords, or synonyms. The framework that is used for matching starts with a superset of keywords that are associated with cloud computing. The superset of keywords was sourced from Amazon, Google and IBM cloud job descriptions as well as from practitioners in the cloud industry including myself who spent approximately 3 years working for a cloud team in the Office of the CIO at IBM.

<u>Amazon:</u>

This section provides a visual of a Black-box ATS interface that is capable of processing 1 resume at a time. I use the Black-box ATS interface here to help the reader visualized resume and job description matching that takes place in batch mode when the Python Analyzer (PA) is executed. The PA reads in all resumes in a folder, processes each and output statistics on matching before and after synonyms are used in the matching process.

For Amazon, Figure 27 below shows a Black-box ATS (BB ATS) interface that takes in a resume (on the left) and a job description (on the right) in an attempt to find a match percentage based on keywords.

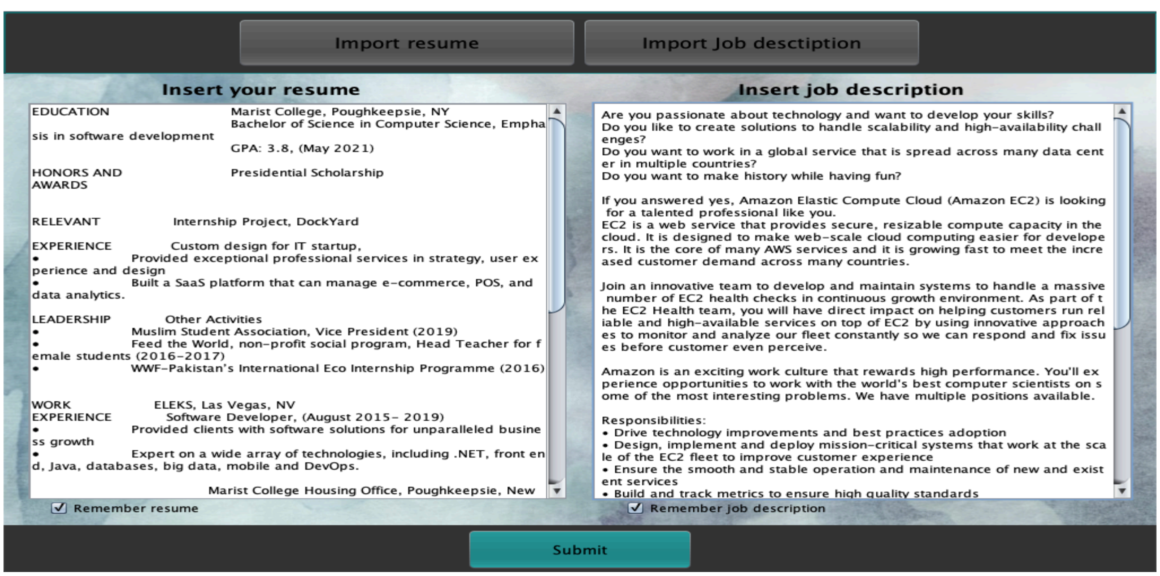

Figure 27 Black-box ATS Interface - AmazonFigure 27

The BB ATS interface takes in a resume (e.g., Tariq_Amazon #1 Resume Team _4_Day) which is describe in appendix A that has approximately 58 keywords that match the superset of keywords and tries to find a match between job description and resume. In comparing keywords in the resume with keywords in the job description there is a 22% match on keywords – see Figure 28.

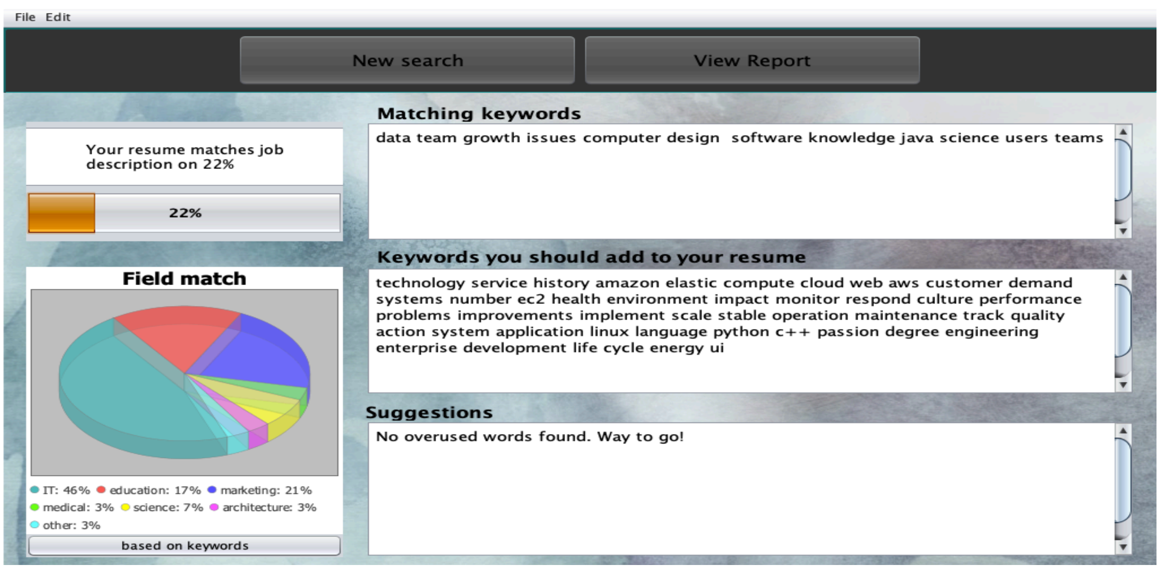

Figure 28 Black-box ATS Interface – Amazon First PassFigure 28

The example above shows a match on keywords "Java and Engineering" which are in the CSD job description for Amazon.

In contrast to the BB ATS, the PA reads in a text file that contains a superset of keywords that have been derived from the Knowledge Graph (KG) and other sources. The PA compares the text file against the CSD job description for Amazon. The output from this comparison results in a common set of keywords that are shared between the text file and job description. The common set of keywords are then compared against resumes from each candidate applying for the CSD job. Statistics are then displayed detailing keyword match percentages before and after synonyms are used in the matching process. The PA calls an API to the Datamuse open-source database and uses up to 5 synonyms for each unmatched keyword after the first try at a match. This effort results improved match percentages.

Google:

This section provides a visual of a Black-box ATS interface that is capable of processing 1 resume at a time. I use the Black-box ATS interface here to help the reader visualized resume and job description matching that takes place in batch mode when the Python Analyzer (PA) is executed. The PA reads in all resumes in a folder, processes each and output statistics on matching before and after synonyms are used in the matching process.

For Google, Figure 29 below shows a Black-box ATS (BB ATS) interface that takes in a resume (on the left) and a job description (on the right) in an attempt to find a match percentage based on keywords.

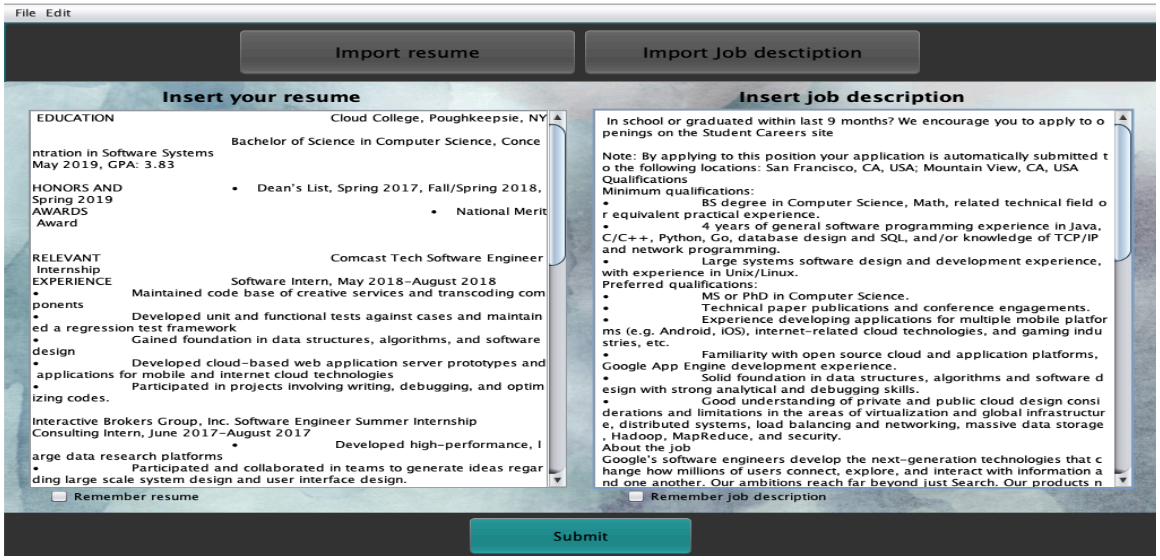

Figure 29 Resume Analyzer Interface - GoogleFigure 29

The BB ATS interface takes in a resume (e.g., 39-Ebeling_Google#2 Resume Team _3_Night) which is describe in Appendix A2 that has approximately 87 keywords that match the superset of keywords and tries to find a match between job description and resume. In comparing keywords in the resume with keywords in the job description there is a 44% match on keywords – see Figure 30.

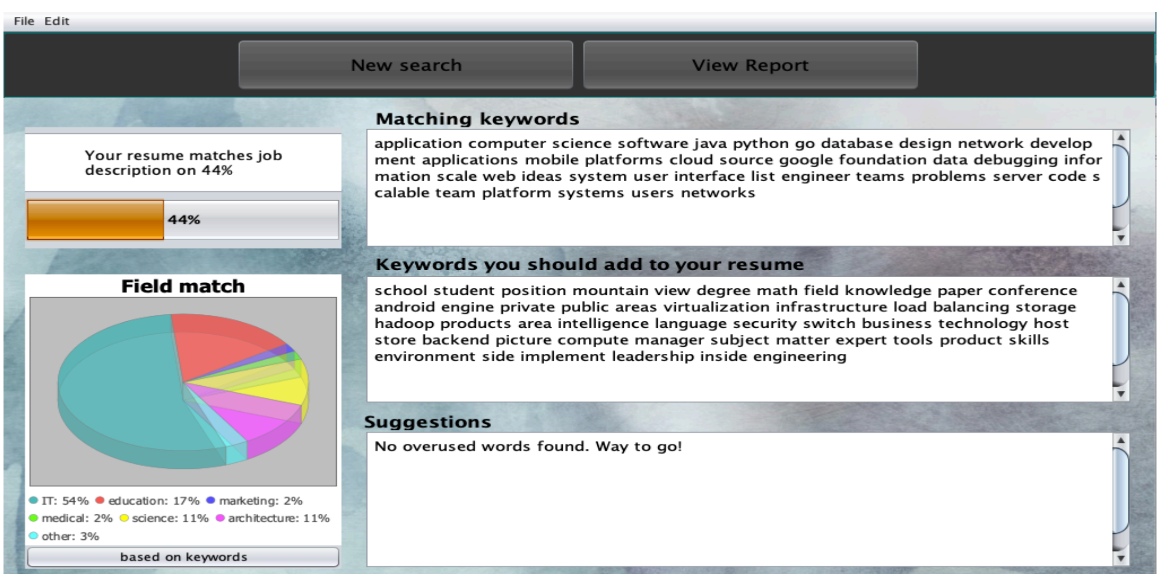

Figure 30 Resume Analyzer Interface – Google First PassFigure 30

The example above shows a match on keywords like "Python and Google" which are in the CSD job description for Google.

In contrast to the BB ATS, the PA reads in a text file that contains a superset of keywords that have been derived from the Knowledge Graph (KG) and other sources. The PA compares the text file against the CSD job description for Google. The output from this comparison results in a common set of keywords that are shared between the text file and job description. The common set of keywords are then compared against resumes from each candidate applying for the CSD job. Statistics are then displayed detailing keyword match percentages before and after synonyms are used in the matching process. The PA calls an API to the Datamuse open-source database and uses up to 5 synonyms for each unmatched keyword after the first try at a match. This effort results improved match percentages.

<u>IBM:</u>

This section provides a visual of a Black-box ATS interface that is capable of processing 1 resume at a time. I use the Black-box ATS interface here to help the reader visualized resume and job description matching that takes place in batch mode when the Python Analyzer (PA) is executed. The PA reads in all resumes in a folder, processes each and output statistics on matching before and after synonyms are used in the matching process.

For IBM, Figure 31 below shows a Black-box ATS (BB ATS) interface that takes in a resume (on the left) and a job description (on the right) in an attempt to find a match percentage based on keywords.

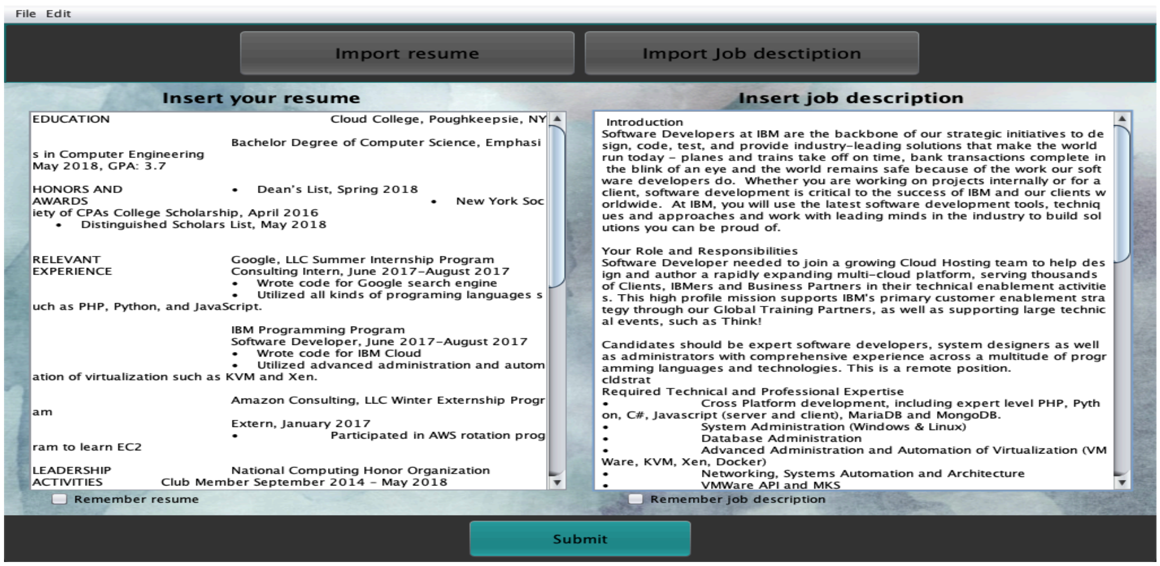

Figure 31 Resume Analyzer Interface – IBMFigure 31

The BB ATS interface takes in a resume (e.g., 35-Jacky_IBM resume_Team _5_Day) which is describe in appendix A3 that has approximately 73 keywords that match the superset of keywords and tries to find a match between job description and resume. In comparing keywords in the resume with keywords in the job description there is a 25% match on keywords – see Figure 32.

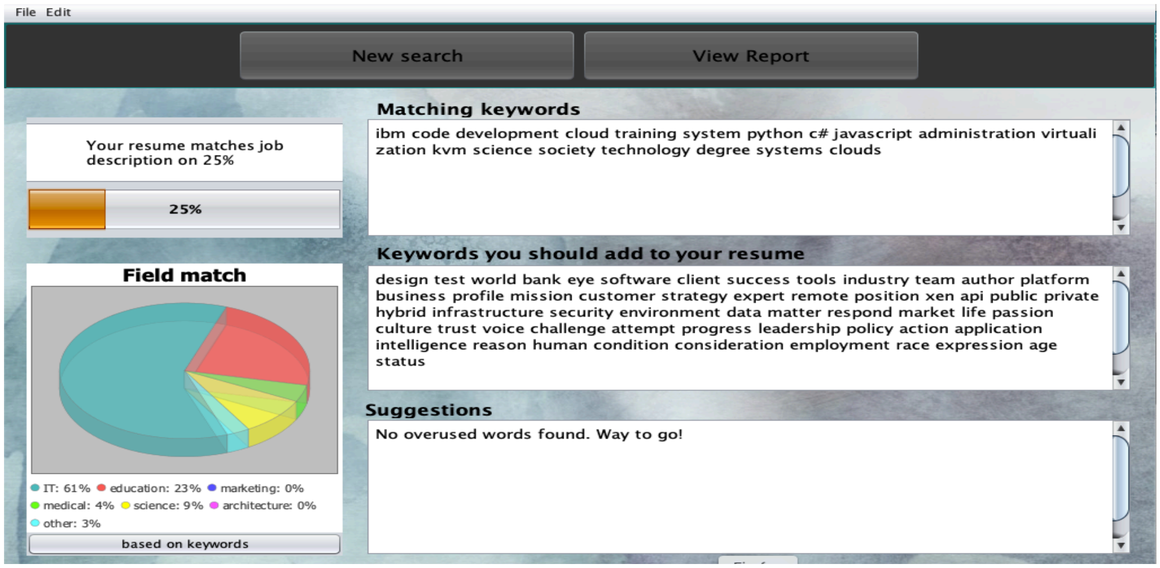

Figure 32 Resume Analyzer Interface – IBM First PassFigure 32

The example above shows a match on keywords like “javascript and kvm” which are in the CSD job description for IBM.

In contrast to the BB ATS, the PA reads in a text file that contains a superset of keywords that have been derived from the Knowledge Graph (KG) and other sources. The PA compares the text file against the CSD job description for IBM. The output from this comparison results in a common set of keywords that are shared between the text file and job description. The common set of keywords are then compared against resumes from each candidate applying for the CSD job. Statistics are then displayed detailing keyword match percentages before and after synonyms are used in the matching process. The PA calls an API to the Datamuse open-source database and uses up to 5 synonyms for each unmatched keyword after the first try at a match. This effort results improved match percentages.

Section Summary:

Examples in this section show that the framework results in improved math percentage between a resume and a job description. In at least one case, the match percentage improvement was substantial enough for the resume to make it to the top 5 list of best candidates. While the framework results in improve matches, additional improvement through custom filters can lead to further refinement of the short list of candidates. For keywords that were not matched, the KG can be revisited to assess whether other important filters (i.e gender, racial status, etc.) need to be considered. When all factors above a considered, job application pre-processing is improved as a result of the framework.

3.1.5 Framework Enhancements Through Other Filters

Combining the KG, PA and Gender to improve matches:

Since the KG is a visual representation of HR requirements, it can be quickly referenced to see whether gender or some other diversity initiative is a priority in the recruiting process. It is possible that the firm may already have a substantial number of white males and would give higher priority to a candidate who is a woman, military veteran or some other member of a historically underrepresented group. "…organizations can enact key activities to attract and retain women in all ranks of the organization …" [11] The PA could help by reading in a new text file that has a superset of keywords that includes a word like "female." This new word will then be used in the keyword and keyword synonym search through resumes to produce a short list of candidates who meet the gender diversity requirements. This ability is especially helpful in male dominated fields like High Tech.

More can be done in this area, but some companies are using features like that to find diverse candidates: “Textio is a Seattle-based startup that develops software to get around this problem by analyzing the text of job postings to make postings attractive to a diverse pool of applicants …” [12]

Combining the KG, PA and Location to improve matches:

The KG can also be quickly referenced to see whether location is a priority in the recruiting process. It is possible that firm may be practicing Agile software development and require employees to be co-located in a specific geography. When I worked in the office of the CIO at IBM, the company embraced a co-location strategy and would only hire employees who could work in strategic locations. If a company adopts such a strategy, then location would be given a higher priority than other matching factors. The PA could help by reading in a new text file that has a superset of keyword that includes a word like “Pleasantville, Boston, or Seattle.” These new words will then be used in the keyword and keyword synonym search through resumes to produce a short list of candidates who meet the location requirements.

Colocation is very important in Agile: “One of the primary ideas behind the foundation of the Agile Methodology was and continues to be Agile team co-location. We have been told, and we tell others, that face-to-face, co-located teams are the optimal goal of Agile and that Business Development, testers, SMEs, and developers must work together daily throughout the project. This is accepted as essential in Agile and is outlined in Principles 4 & 6, around which the Agile Manifesto is based: ‘Close, daily cooperation between business people and developers' and 'Face-to-face conversation is the best form of communication [co-location].'" [13]

Combining the KG, PA and Field/Specialty Skill area to improve matches:

Finally, the visual nature of the KG can be referenced to see whether a particular candidate has Field/Specialty Skill expertise that meets the requirement of the jobs. When the "Submit" button of the BB ATS is pressed, in addition to comparing keywords in the resume and job description, resume matches with job fields are captured – see Figure 33 below.

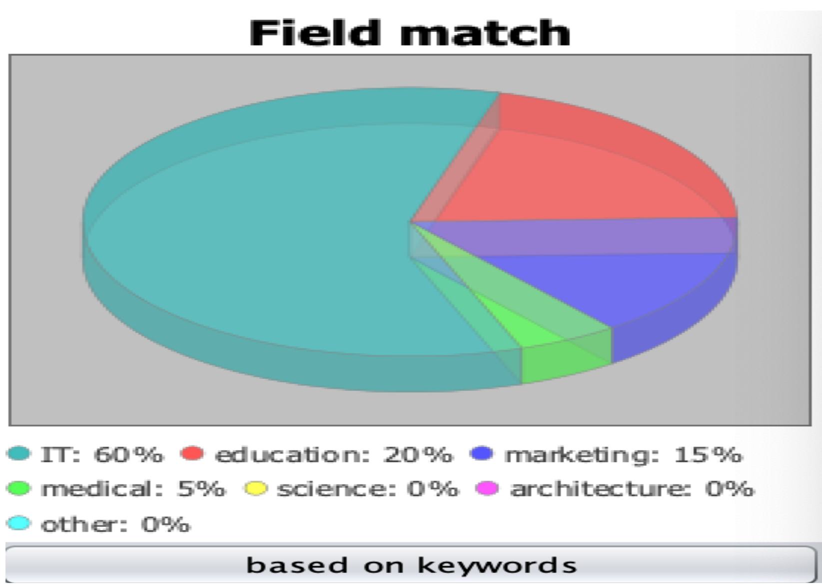

Figure 33 BB ATS Field MatchingFigure 33

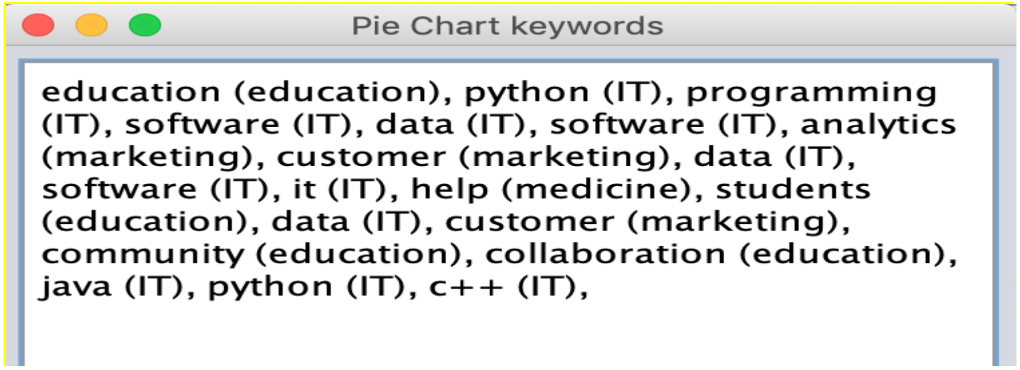

Figure 34 Keywords used in "Field Matching" ChartFigure 34

Minor adjustment can be made to the PA to enable similar function. This could be a valuable feature as many jobs require expertise in multiple areas. By reviewing percentages in Figure 33, HR personnel can see if a candidate has skills that fall under multiple disciplines: IT, Education, Marketing and Medical. If having knowledge in multiple areas is important, then the "Field Matching" feature could be given a higher priority than other factors. HR can then select candidates who match skill areas of importance. To assess strengths in skills areas, some candidates like those that have IT skills could be asked to take a pre-employment/coding test through a company like Hacker-Rink to assess coding skills. "A preemployment test is an objective and standardized device used to gauge a person's KSAO relative to order individuals." [19] KSAO is an abbreviation for Knowledge, Skill, Abilities and Other features that measures a candidate's competencies.

3.2 Applying the Framework to Industries

3.2.1 Medical Industry

The Framework described above can also be applied to the medical industry. Research indicates that keyword searches have been use to find relationship among rheumatic diseases. “The MEDLINE database (Medical Literature Analysis and Retrieval System Online) contains an enormously increasing volume of biomedical articles. There is urgent need for techniques which enable the discovery, the extraction, the integration and the use of hidden knowledge in those articles.” [15] I believe that the PA framework can be slightly modified to help find matches among rheumatic diseases describe in the MEDLINE database.

Similarly, the PA framework could be used to help fight cancer. Consider a scenario where a patient has glioblastoma brain cancer, and doctors want to develop a short list of treatment options for the patient. Doctors could create a list of keywords based on chemotherapy terms and other treatment option terms (i.e., DNA marker names, etc.) that have been used to successfully treat cancers in the past. A KG can then be developed to visualize relationship between keyword terms that have been successful in glioblastoma treatment in the past. These keywords can then be loaded into the PA and compared to terms in a patient’s medical chart to determine the degree to which terms used previously might match keywords in a client’s medical charts. If there are matches, successful treatment options could be considered for the patient.

A KG similar to the one in Figure 35 could also be decomposed leading to more detailed visuals of relationships between successful treatment keywords and patient information.

We could then use KG keywords and synonyms to search for matches in journal article, clinical studies and cancer databases to see if additional treatment options are possible.

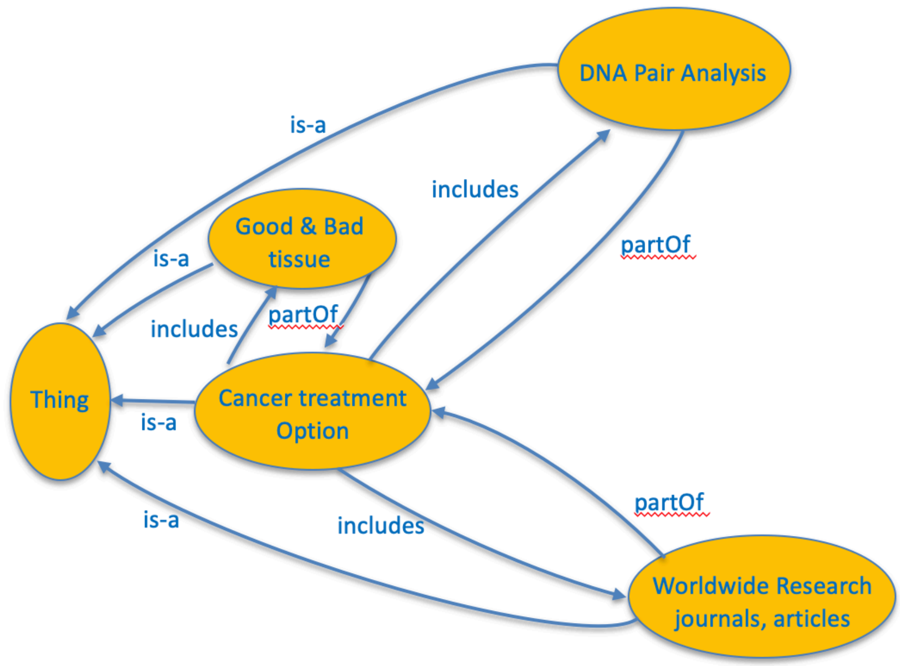

Figure 35 PA Framework and the Medical IndustryFigure 35

In the KG above, Cancer Treatment Option is a thing and includes other things like Good or Bad tissue (i.e., cancerous/non-cancerous tissue), a patient's DNA information as well as information from written sources (i.e., journals, etc.). Some terms from the KG can be feed into the PA and used for matching between treatment options and tissue mutation, etc.

3.2.2 Academic Industry

The Framework described above can also be used to find the most qualified students to admit to an academic institution. "We note that the problem of selecting a few candidates from a large pool arises not only in job hiring but also in many other fields such as college admissions settings" [7] . Consider a scenario where a top College (e.g., Pace) has to select the top 10% of students who apply. We could create a list of keywords based on standardized test scores, recommendations and/or High School graduation status to construct a KG similar to Figure 36 below to see relationships among keywords. We would then compare KG keywords and synonyms in admission standards to keywords found in student profiles. A short list of candidates would result from the comparison that could be used in assessing which students are in the top 10 – 20 % of students.

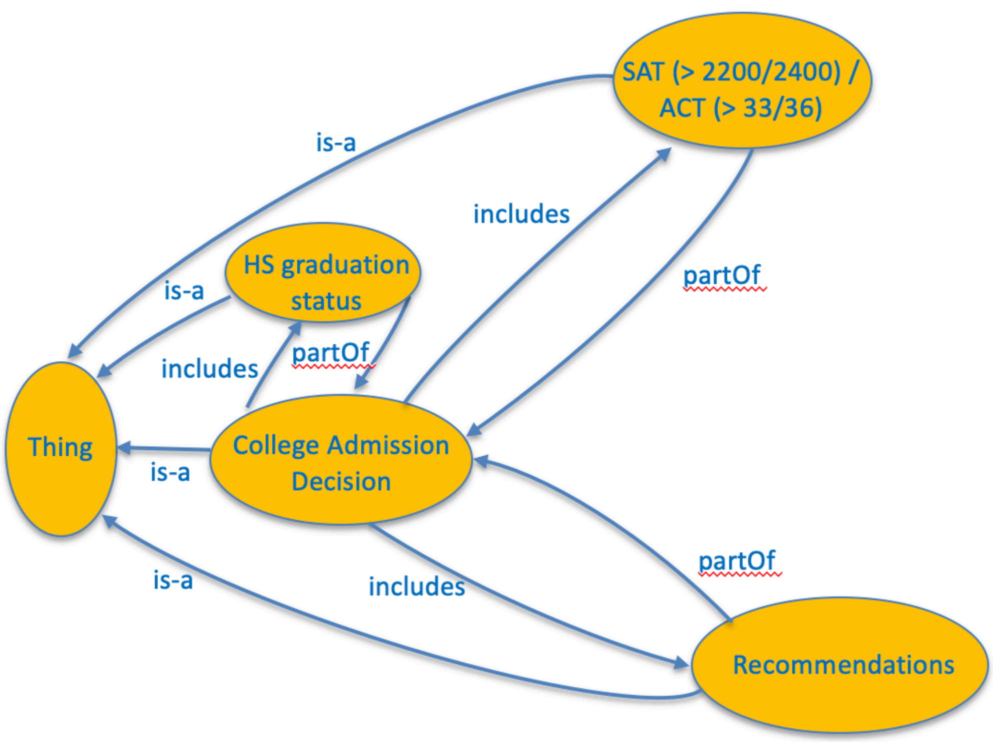

Figure 36 RA Framework and the Academic IndustryFigure 36

In the KG above, a College Admission Decision is a thing and includes other things like High School graduation status, SAT scores, and letters of recommendation. Some terms from the KG can be feed into the PA and used for matching between Admission decision criteria and SAT scores, etc.

3.2.3 Retail Industry

Retail stores that issue loyalty cards often track product sales for such items as cereal and/or milk. This data can also be used to analyze how often products are sold together. By using the PA Framework, keywords like product/SKU number can be used to determined how often 2 or more SKU numbers are used together. By matching sales of one SKU to sales with sales of another SKU, retailers can understand user preferences for one product when buying another. This can influence advertising decision as well as decisions about cross-selling or up-selling products. “Therefore, the customer's search behavior can influence the buying behavior of the customer, and it can be used for recommending products to a similar customer based on the purchase history of the customer whose search keywords are similar.” [33]

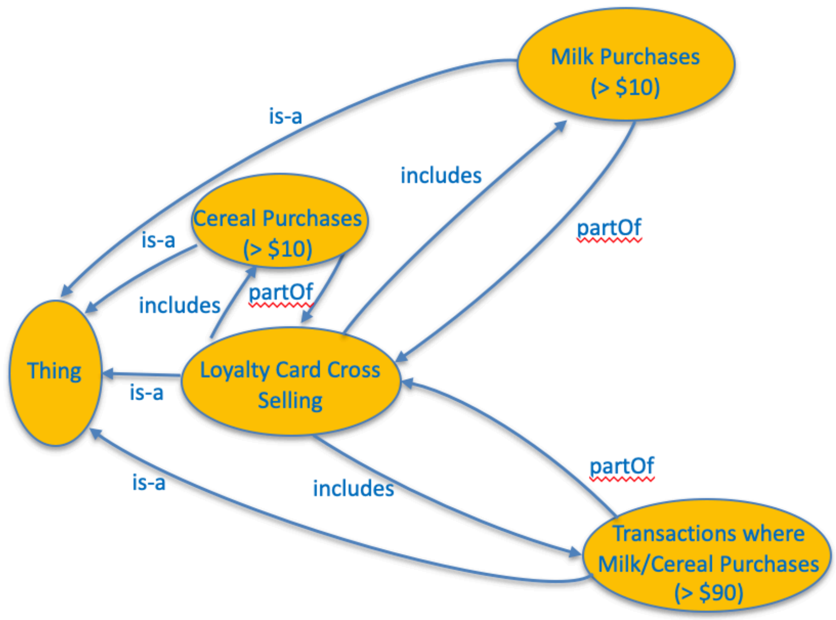

Figure 37 RA Framework and the Retail IndustryFigure 37

In the KG above, a Loyalty Card Cross Selling is a thing and includes other things like products purchased (i.e., milk) or transactions (i.e., multiple purchases where milk and cereal are purchased together). Some terms from the KG can be feed into the PA and used for matching between products that are sold separately and products that could be sold together (i.e., cross/up selling).

3.2.4 Banking Industry

Banking – Banks can use the PA Framework by modeling a decision tree that assigns mortgages to an approval department based on keyword matching that indicates which loans are likely to result in a approval or rejection/default. A similar approach was done with a decision tree when assigning IT tickets to the correct Help-Desk group for assistance: "The model endeavors to match the most appropriate resolver group with the type of the incident ticket on behalf of the IT service desk function." [25] For banks that lend money, a KG like the one below can be created from a decision tree. Keywords from the decision tree can be used to a create relationships associated with a customer's employment status, loan payment history or loan-to-value ratios. Using this information, bankers can make decisions regarding whether to approve or deny a requested loan.

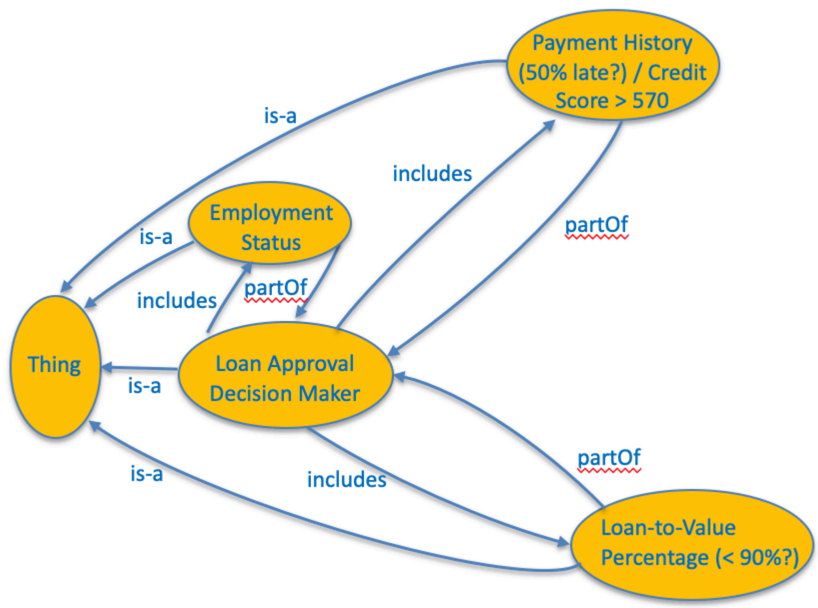

Figure 38 RA Framework and the Banking IndustryFigure 38

In the KG above, a Loan Approval Decision Maker is a thing and includes information to allow decision making based on employment status, payment history, etc. Some terms from the KG can be feed into the PA and used for matching to determine which loan met the criteria for approval.

Summary:

This chapter starts out by introducing the Python Analyzer (PA) Framework that can be used across multiple industries. A Black-box ATS (BB ATS) is highlighted in this section to provide a visual of the matching process between a text resume and text job description. This BB ATS process has a limitation as to the number of resumes and job description that can be processed at one time. To that end, 1 resume at a time is feed into the BB ATS and we are able to visualize matching results from 3 resumes and 3 job descriptions from Amazon, Google and IBM. In contrast, the PA ATS that is used in the PA Framwork will read an entire folder of resumes and display results accordingly. The PA Framework also allows keywords to be gather from various sources and a KG is constructed from keywords showing important important relationships that may not be capable of modeling in ontology tools. Based on relationship in the KG, a base set of keywords are loaded into the PA and the matching process begins. An overview of how PA Framework can be applied to improve matching in the Medical , Academic, Retail and Banking industries is also provided.

Chapter 4 Experiment Evaluation

This chapter describes how a software program called Python Analyzer (PA) was used to help test matches between resumes and job descriptions. Section 4.1 provides a clear description of the experiment design and process as well as highlights sources of data (i.e., Subject Matter Experts, etc.) used to complete this research. Findings in this section are also corroborated by SMEs and statistical regression. A description of how resumes and job descriptions are used in the Job Application Pre-processing (JAP) evaluation process is also provided. Section 4.2 highlights differences between the traditional JAP approach and the PA approach. Section 4.4 summarizes the chapter.

4.1 Experiment Design and Process

<u>Experiment Design:</u>

The research experiment was designed to be quantitative in order to assess whether improvements in match percentage were observed following the experiment.

The experiment design also requires the creation of a computer program to evaluate matches on keywords between a job description and resumes. To that end, a Python program called Python Analyzer (PA - PythonAnalyzer20.py) was created to support the experiment. Section 3.1.1 has a flowchart and pseudo code for the PA. Appendix D has the actual source code.

<u>Experiment Inputs:</u>

The PA program reads in the following files as input:

- **Knowledge Graph keywords file:** a text file titled "keywords.txt" that was created based on knowledge graphs and analysis of information from SMEs and Cloud Software Developer job descriptions is needed for the PA. This file should be placed in the main directory path from which the PA program will run/executed
- **Job Description files:** three Job Description text titled data1.txt, data2.txt and data3.txt are needed for the PA. The data1.txt file contains a job description for an IBM cloud job; The data2.txt file contains a job description for an Amazon cloud job; and the data3.txt file contains a job description for a Google cloud job. These files should be place in a folder called "Data" which should exist in a user's main directory path
- **Resume files:** three folders data1, data2 and data3 will need to be created off of the user's main directory path. The data1 folder should be populated with resumes to be used when processing matches for the IBM job description referenced above (i.e., data1.txt). The data2 folder should be populated with resumes to be used when processing matches for the Amazon job description (i.e., data2.txt). The data3 folder should be populated with resumes to be used when processing matches for the Google job description (i.e., data3.txt). Resumes were received in multiple formats (.docx, .pdf, etc.). The PA will only accept .docx resume files for now and converts these files to .txt files during processing.
- Before running the PA program, a user should make sure that Python environment has the following libraries if not already installed: os, docx2txt, glob, pandas as pd,

string, and from datamuse import datamuse. The PA program will import these libraries.

Data Collection Process for Resumes:

Approximately 120 resumes from college students broken into 10 teams were used as input to the PA for this research – see Table 5 for more detail. To avoid collecting student personal identifiable information, students labeled resumes were using a combination of company name and team number (i.e., Google#2 Resume Team _3_Night) – see Appendices A, B and C for examples. During the match comparison of word from resumes to keywords from job descriptions, only a text-to-text comparison is made.

Participants in the matching evaluation process included 10 teams of approximately 5 students who provided 12 resumes - 4 for each of the 3 job descriptions from Amazon, IBM and Google. Student names are not included on resumes and a text-to-text analysis is conducted to assess a match between resume text and job description text. A breakdown of how the 120 resumes were collected is described in the matrix below.

Table 6 List of Resumes by Company and by Student TeamTable 6

Teams\ Company	Team 1 Day class \ Team 1 Night class	Team 2 Day class \ Team 2 Night class	Team 3 Day class \ Team 3 Night class	Team 4 Day class \ Team 4 Night class	Team 5 Day class \ Team 5 Night class	Total Resumes
No. Of Resumes for Amazon	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	40
No. Of Resumes for IBM	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	40
No. Of Resumes for Google	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	4 \ 4 = 8	40
Total Resumes	24	24	24	24	24	120

Experiment Process:

The PA computer program reads in input files described above including resumes created for each of the 3 cloud job descriptions from IBM, Amazon and Google. Words from the Knowledge Graph Keywords file are assessed against the job description word to create a list of keywords to be used for assessing matches with words in each resume. When a match is found, a count is tallied. Synonyms for keywords not match are then used to see if more matches against resumes are possible. If appropriate, match totals are update and statistics are prepared for printing or displaying on the screen. The flowchart and pseudo code in from section 3.1 and 3.1.1 has a more details on the PA algorithm.

Experiment Output:

Section 4.3 has tables, graphs and statistics that highlight output and observations from this research. For each resume processed associated with a job description, the PA produced match percentages before and after synonyms were used to help improve matches. The output also displays the top 10 resumes with the highest match percentages. PA output shows that the PA algorithm results in match improvements.

A cloud Subject Matter Experts (SME) reviewed resumes for each job description and selected many of the same resumes as the PA for his top 3 to 5 candidates.

To corroborate output finds, a regression analysis was run to see if results were due to chance. The R-Squared and p-value indicates that differences are statistically significant.

4.2 Traditional Versus Python Analyzer Approach

4.2.1 Traditional Recruiting Method

Traditional Job Application Pre-processing (JAP) involves having an Applicant Tracking System (ATS) or bot software filter through several hundred resumes trying to find a match on keywords between words in a resume and word in a job description. A short list of resumes with the highest match relative to words in the job description is produced and this subset of resumes are considered to be the best fit or most qualified candidates. Human

Resources (HR) personnel will then sort through the short list of resumes trying to further reduce the shortlist based on other criteria like keywords that match college/schools of interest, a state or location of interest or specific graduation date that has been specified in the ATS. Eventually, a list of candidates is shared with hiring managers and these managers meet with team leaders or other managers to determine which candidates should be interviewed.

Since HR personnel are often not experts in the area for which they are recruiting and the ATS is not always accurate, some qualified candidates are missed, overlooked or never make it to the short list that get reviewed by managers and leaders.

4.2.2 Python Analyzer Pre-processing Method

The Python Analyzer (RA) was developed to mimic commercial ATS software, but also enhanced to match on keyword synonyms. This allows HR personnel (HRP) to replace the traditional ATS and/or use the traditional ATS as a first filter, then run PA in consultation with Knowledge Graphs to reduce the number of resumes in the ATS short list by applying further filters.

With PA, HRP are able to feed a list of resumes into the PA and obtain a match percentage before and after synonyms matches are applied. In the end, a short list of the best candidates can be obtained and analyzed.

The PA approach eliminates the following pitfalls in traditional methods and practices:

1. eliminates the process of conducting multiple interviews over several weeks where HRP interviews candidates for an initial screening and have a technical person interview candidates again to clarify qualifications (i.e. can you write computer code in Java?)
2. eliminates spending money for candidates to visit a company unless the candidate is amongst the top 3 to 5 job prospects
3. eliminates expensive testing (i.e., code academy, triple-byte programmers test) for everyone and focus on just top candidates
4. eliminates or minimized the level of domain-specific knowledge required for recruiters to screen candidates during recruiting

4.3 Experiment Results

This section describes how results are validated for the approximately 120 resumes and the 3 job descriptions. A cloud job description for Amazon, Google and IBM will be used in our experiment. Tables 7, 8 and 9 contain a row for each resume showing match percentages before and after synonyms are used to find a match with the respective job description being compared with the resume. For additional comparison purposes, in the first column of the table, is an average match percentage of all resumes processed using a BB ATS software program. The last column shows match difference between the PA ATS and the BB ATS. The data shows that there is an improvement in match percentages keyword synonyms are included to improve matches between resumes and job descriptions.

To corroborate finds, a regression analysis was run to see if results in this section a due to chance. Finally, I have solicited input from practitioners to see if they would select the same resumes for their top 5 as the PA selected for its top 5.

4.3.1 Analysis of Amazon Match Results

In Table 7 below, the numbers under the "PA Match" column show the match percentage of each resume before synonyms are introduced. Numbers under the "PA Match with Synonyms" column show the match percentage of each resume after introducing matching based on keyword synonyms. The Blackbox ATS (BB ATS) column shows the average match percentage for all resumes using a BB ATS solution.

Table 7 Results from Amazon Resume and Job Description MatchingTable 7

Resume Number	Black-Box ATS	PA Match	PA Match with Synonym	Change from Black-Box	Change from PA Match	% Change from PA Match
1-Mmandile Amazon#1 Resume Team _1_Night	20.1	45	45	24.9	0	0.00%
2-Peck_Amazon#1 Resume Team _1_Night	20.1	20	25	4.9	5	25.00%
3-Larsen_Amazon#1 Resume Team _1_Night	20.1	45	45	24.9	0	0.00%
4-Currier_Amazon#1 Resume Team _2_Night	20.1	15	15	-5.1	0	0.00%
5-Glynn_Amazon#1 Resume Team _2_Night	20.1	20	25	4.9	5	25.00%
6-Ebeling_Amazon#1 Resume Team _3_Night	20.1	20	25	4.9	5	25.00%

7-Doyon_Amazon#1 Resume Team _5_Night	20.1	20	20	-0.1	0	0.00%
8-Mazut_Amazon#1 Resume Team _1_Day	20.1	30	30	9.9	0	0.00%
9-Regan_Amazon#1 Resume Team _2_Day	20.1	35	40	19.9	5	14.29%
10-Regan_Amazon#2 Resume Team _2_Day	20.1	20	25	4.9	5	25.00%
11-Regan_Amazon#3 Resume Team _2_Day	20.1	25	35	14.9	10	40.00%
12-Regan_Amazon#4 Resume Team _2_Day	20.1	25	35	14.9	10	40.00%
13-Logue_Amazon #1 Resume Team _3_Day	20.1	30	40	19.9	10	33.33%
14-Logue_Amazon #2 Resume Team _3_Day	20.1	20	25	4.9	5	25.00%
15-Logue_Amazon #3 Resume Team _3_Day	20.1	35	40	19.9	5	14.29%
16-Logue_Amazon #4 Resume Team _3_Day	20.1	30	30	9.9	0	0.00%
17-Boley_Amazon #1 Resume Team _4_Day	20.1	20	25	4.9	5	25.00%
18-Kiss_Amazon #1 Resume Team _4_Day	20.1	40	45	24.9	5	12.50%
19-Tariq_Amazon #1 Resume Team _4_Day	20.1	25	40	19.9	15	60.00%
20-DiChiara_Amazon#1 Resume Team _5_Night	20.1	45	45	24.9	0	0.00%
21-Herrera_Amazon#1 Resume Team _4_Night	20.1	30	40	19.9	10	33.33%
22-Sumi_Amazon#1 Resume Team _4_Night	20.1	45	60	39.9	15	33.33%
23-Attala_Amazon#1 Resume Team _4_Night	20.1	30	35	14.9	5	16.67%
24-Ebeling_Amazon#2 Resume Team _3_Night	20.1	45	50	29.9	5	11.11%

25-Ebeling_Amazon#3 Resume Team _3_Night	20.1	25	30	9.9	5	20.00%
26-Ebeling_Amazon#4 Resume Team _3_Night	20.1	40	45	24.9	5	12.50%
27-Monono_AmazonI#1 Resume Team _2_Night	20.1	15	20	-0.1	5	33.33%
28-Viri_Amazon#1 Resume Team _5_Night	20.1	40	45	24.9	5	12.50%
29-Palmer_Amazon#1 Resume Team _1_Day	20.1	30	35	14.9	5	16.67%
30-Kati_Amazon Resume#2_Team _5_Day	20.1	30	35	14.9	5	16.67%
31-Kati_Amazon Resume#1_Team _5_Day	20.1	15	25	4.9	10	66.67%
32-Jacky_Amazon Resume#1_Team _5_Day	20.1	35	40	19.9	5	14.29%
33-Jacky_Amazon resume#2_Team _5_Day	20.1	40	45	24.9	5	12.50%
Average	20.1	29.8	35.2	15.1	5.3	17.77%

From a summary point of view, 33 Amazon resumes were feed through the Python Analyzer (PA) and matches between the PA with keyword synonym matching and a BB ATS solution varied from – 0.1% to 29.9% with an average change of a little more than 15.1%.

If we examine improvements as a percentage of the initial PA match on keywords alone, we see that algorithm adjustments to introduce matching based synonyms made a significant difference. In fact, at least 24% (8/33) of resumes showed improvements over 30% with the biggest improvement being 66.6% (from a 15% to 25% for 1 resume).

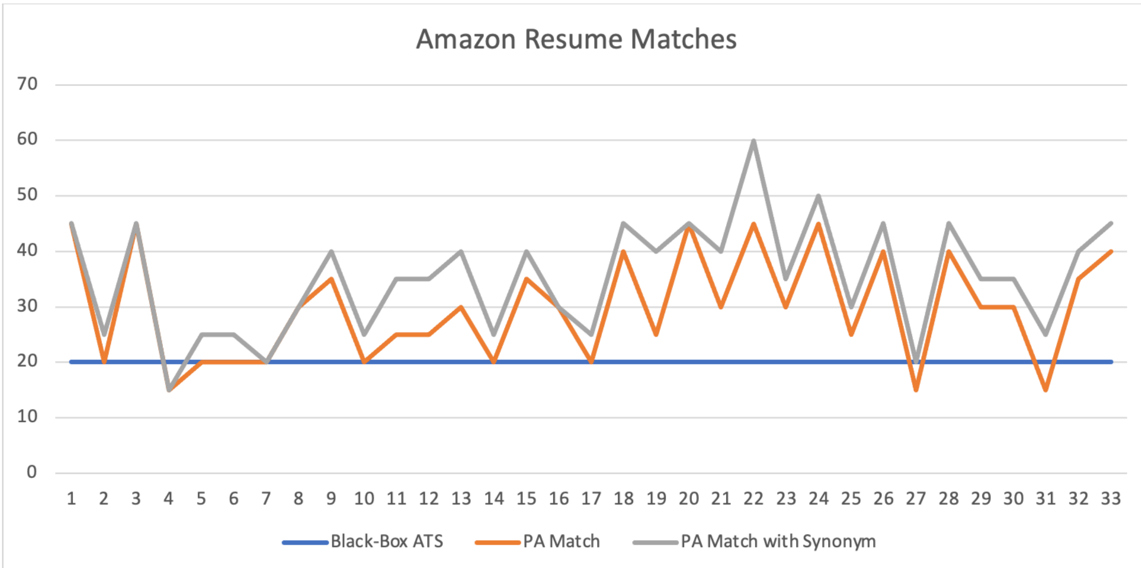

Figure 39 Line Graph of Resume Matches for AmazonFigure 39

When calculating the difference between PA Match and PA Match with Synonym in Table 7, I used the following:

Formula = (PA Match with Synonym – PA Match)

As you can see from the graph, the line for PA Match with Synonym is higher for almost all of the resumes. This means that matching is likely to improve whenever synonyms and other filters are involved. The PA Match with Synonym line is also higher than the line that represents the average match percent for Black-Box ATS matching.

<u>Regression Analysis</u>

A review of Regression Table 8 below, indicates an R Square value of .852, which means that the predictor variable in the model (PA Match) explains 85.2% (.852x100) of the variance in the outcome variable PA Match With Synonyms.

Also note that the second column in the table contains coefficients and the fifth column contains p values associated with the regression. The coefficient for PA Match is .960 and has a p-value of .000 which is less than the cutoff (alpha level) of .05. The p-value means that this relation is statistically significant. Thus, the more that the matching algorithm makes use of synonyms, the more likely we are likely to see improvements in match percentage.

Table 8 Regression Results from Amazon Resume and Job Description MatchingTable 8

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	.923							
R Square	.852							
Adjusted R Square	.847							
Standard Error	3.984							
Observations	33.000							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	1.000	2832.233	2832.233	178.450	.000			
Residual	31.000	492.010	15.871					
Total	32.000	3324.242						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	6.502	2.254	2.885	.007	1.905	11.099	1.905	11.099
PA Match	.960	.072	13.359	.000	.813	1.106	.813	1.106

Subject Matter Expert Analysis

I asked an IT professional who is Amazon Web Service (AWS) certified to review 33 resumes for the Amazon Cloud Software Developer's job and list the top 5 candidates most qualified for the job. Below is a list of the top 5 resumes that the IT professional provided:

Top 5 (no particular order):

1. Team 3: Amazon Resume #4 (Ebeling_Amazon#4 Resume Team _3_Night.docx)
2. Team #4 BUS 301 – Amazon Resume (Palmer_Amazon#1 Resume Team _1_Day.docx)
3. Team#4_115-Amazon_Resume#4 (Herrera_Amazon#1 Resume Team _4_Night.docx)
4. Alec Larsen Amazon Resume (Larsen_Amazon#1 Resume Team _1_Night.docx)
5. Team#5_201-Amazon_Resume#1 (Sumi_Amazon#1 Resume Team _4_Night.docx)

The Python Analyzer's top 5 are listed below in order:

1. 22-Sumi_Amazon#1 Resume Team _4_Night – 60% - - → SME's #5
2. 24-Ebeling_Amazon#2 Resume Team _3_Night – 50%
3. 3-Larsen_Amazon#1 Resume Team _1_Night – 45% - - → SME's #4
4. 26-Ebeling_Amazon#4 Resume Team _3_Night – 45% - - → SME's #1
5. 33-Jacky_Amazon resume#2_Team _5_Day – 45%

Three of the SME's candidates appears in the PA's top 5 list of candidates. This means that both groups thought that 85% of resumes were not the best match, and that of the remaining 15%, there is agreement on 60% (3 of 5) of candidates – see highlighted font above for details.

4.3.2 Analysis of Google Match Results

In Table 9 below, the numbers under the "PA Match" column show the match percentage of each resume before synonyms are introduced. Numbers under the "PA Match with Synonym" column show the match percentage of each resume after introducing matching based on keyword synonyms. The Blackbox ATS (BB ATS) column shows the average match percentage for all resumes using a BB ATS solution.

Table 9 Results from Google Resume and Job Description Matching

Resume Number	Black-Box ATS	PA Match	PA Match with Synonym	Change from Black-Box	Change from PA Match	% Change from PA Match
2-Lucia_Google#1 Resume Team _1_Night	19	24	27	9	3	12.50%
3-Mmandile Google#1 Resume Team _1_Night	19	24	30	12	6	25.00%
4-Peck_Google#1 Resume Team _1_Night	19	12	18	-1	6	50.00%
5-Currier_Google#1 Resume Team _2_Night	19	3	3	-16	0	0.00%
6-Glynn_Google#1 Resume Team _2_Night	19	3	12	-7	9	300.00%

7-Podesk_Google#1 Resume Team _2_Night	19	1	9	-10	8	809.00%
10-Ebeling_Google#1 Resume Team _3_Night	19	3	9	-10	6	200.00%
11-Elenkov_Google#1 Resume Team _4_Night	19	18	30	12	12	66.67%
12-Elenkov_Google#2 Resume Team _4_Night	19	15	27	9	12	80.00%
13-Elenkov_Google#3 Resume Team _4_Night	19	18	33	15	15	83.33%
14-Doyon_Google#1 Resume Team _5_Night	19	3	9	-10	6	200.00%
15-Lawrie_?Google #2 Resume Team _5_Night	19	18	24	6	6	33.33%
16-Lawrie_?Google #1 Resume Team _5_Night	19	6	9	-10	3	50.00%
17-Hanley_Google#2 Resume Team _1_Day	19	15	24	6	9	60.00%
18-Hanley_Google#1 Resume Team _1_Day	19	12	21	3	9	75.00%
19-Regan_Google#1 Resume Team _2_Day	19	9	15	-4	6	66.67%
20-Regan_Google#2 Resume Team _2_Day	19	12	21	3	9	75.00%
21-Regan_Google#3 Resume Team _2_Day	19	9	21	3	12	133.33%
22-Regan_Google#4 Resume Team _2_Day	19	15	24	6	9	60.00%
23-Logue_Google #1 Resume Team _3_Day	19	30	36	18	6	20.00%
24-Logue_Google #2 Resume Team _3_Day	19	33	33	15	0	0.00%
25-Logue_Goggle #3 Resume Team _3_Day	19	18	27	9	9	50.00%
26-Logue_Google #4 Resume Team _3_Day	19	18	21	3	3	16.67%

27-Boley_Google #1 Resume Team _4_Day	19	12	21	3	9	75.00%
28-Fren Google #1 Resume Team _4_Day	19	15	24	6	9	60.00%
29-Pere_Google #1 Resume Team _4_Day	19	1	3	-16	2	203.00%
30-Tariq_Google #1 Resume Team _4_Day	19	27	36	18	9	33.33%
31-Tariq_Google #2 Resume Team _4_Day	19	27	36	18	9	33.33%
32-Pansini_Google #1 Resume Team _5_Night	19	33	36	18	3	9.09%
33-DiChiara_Google#1 Resume Team _5_Night	19	21	30	12	9	42.86%
34-Herrera_Google#1 Resume Team _4_Night	19	15	27	9	12	80.00%
35-Sumi_Google#1 Resume Team _4_Night	19	9	21	3	12	133.33%
36-Attala_Google#1 Resume Team _4_Night	19	9	18	-1	9	100.00%
37-Ebeling_Google#4 Resume Team _3_Night	19	12	15	-4	3	25.00%
38-Ebeling_Google#3 Resume Team _3_Night	19	9	18	-1	9	102.00%
39-Ebeling_Google#2 Resume Team _3_Night	19	37	42	24	6	15.90%
40-Monono_Google#1 Resume Team _2_Night	19	3	9	-10	6	200.00%
41-Viri _Google#1 Resume Team _5_Night	19	12	18	-1	6	50.00%
42-Palmer_Google#2 Resume Team _1_Day	19	12	18	-1	6	50.00%
43-Palmer_Google#1 Resume Team _1_Day	19	9	18	-1	9	100.00%
44-Alexis_Google #1 Resume Team _5_Day	19	9	15	-4	6	66.67%

45-Alexis_Google #2 Resume Team _5_Day	19	12	15	-4	3	25.00%
46-Derek_Google Resume#1_Team _5_Day	19	9	18	-1	9	100.00%
47-Derek_Google Resume#2_Team_5_Day	19	18	27	9	9	50.00%
Average	19	14	22	3	7	50.68%

From a summary point of view, 44 Google resumes were feed through the Python Analyzer (PA) and matches between the "PA with keyword synonym" and a BB ATS solution varied from – 16% to 24% with an average change of a little more than 3%.

If we examine improvements as a percentage of the initial PA match on keywords alone, we see that algorithm adjustments to introduce matching based synonyms made a significant difference. In fact, at least 27% (12/44) of resumes showed improvements over 100% with the biggest improvement being 809% (from a 1% to 9% for 1 resume).

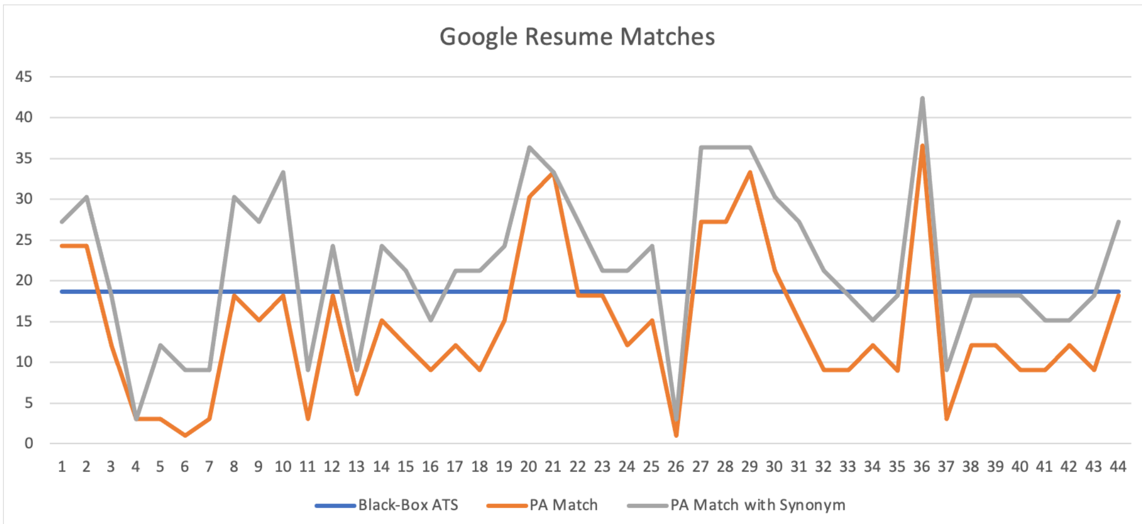

Figure 40 Line Graph of Resume Matches for Google

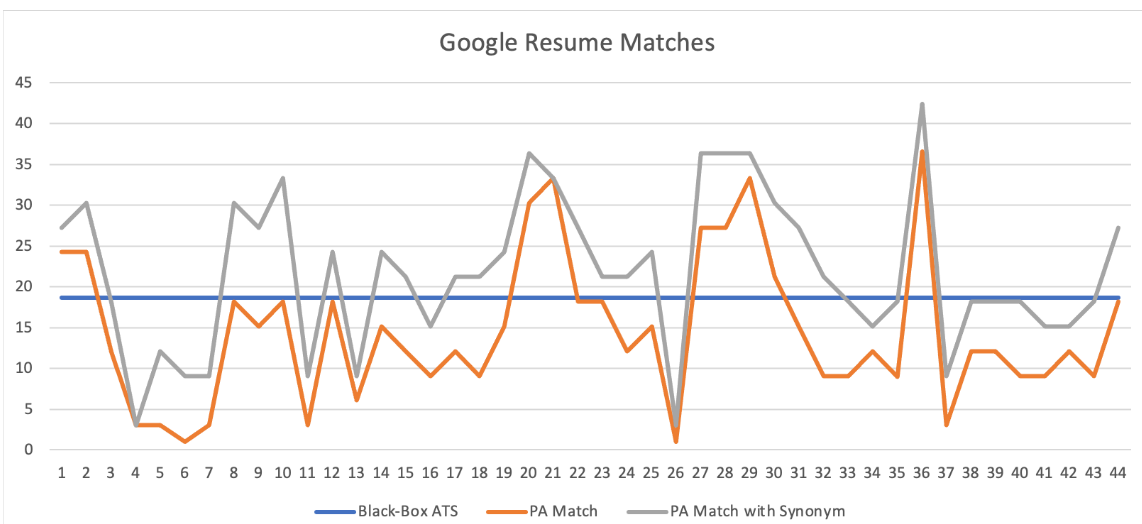

Figure 40

When calculating the difference between PA Match and PA Match with Synonym in Table 9, I used the following:

Formula = (PA Match with Synonym – PA Match)

As you can see from the graph, the line for PA Match with Synonym is higher for resumes. This means that matching is likely to improve whenever synonyms are involved. The PA Match with Synonym line is also higher than the line that represents the average match percent for Black-Box ATS matching most of the time.

Regression Analysis

A review of Regression Table 9 below, indicates an R Square value of .871, which means that the predictor variable in the model (PA Match) explains 87.1% (.871x100) of the variance in the outcome variable PA Match With Synonyms.

Also note that the second column in the table contains coefficients and the fifth column contains p values associated with the regression. The coefficient for PA Match is .978 and has a p-value of .000 which is less than the cutoff (alpha level) of .05. The p-value means that this relation is statistically significant. Thus, the more that the matching algorithm makes use of synonyms, the more likely we are likely to see improvements in match percentage.

Table 10 Regression Results from Google Resume and Job Description MatchingTable 10

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	.933							
R Square	.871							
Adjusted R Square	.868							
Standard Error	3.420							
Observations	44.000							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	1.000	3314.730	3314.730	283.443	.000			
Residual	42.000	491.170	11.695					
Total	43.000	3805.900						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	7.637	.985	7.756	.000	5.650	9.624	5.650	9.624
PA Match	.978	.058	16.836	.000	.861	1.095	.861	1.095

<u>Subject Matter Expert Analysis</u>

I asked an IT professional who is Amazon Web Service (AWS) certified to review 44 resumes for the Google Cloud Software Developer's job and list the top 3 to 5 candidates most qualified for the job. Below is a list of the top 3 resumes that the IT professional provided:

Top 3 (no particular order):

1. Derek_Google Resume#1_Team _5_Day
2. Ebeling_Google#1 Resume Team _3_Night
3. Elenkov_Google#1 Resume Team _4_Night

The **Python Analyzer's** top 5 are listed below in order:

1. 39-Ebeling_Google#2 Resume Team _3_Night– 42%
2. 31-Tariq_Google #2 Resume Team _4_Day – 36%
3. 30-Tariq_Google #1 Resume Team _4_Day – 36%
4. 23-Logue_Google #1 Resume Team _3_Day – 36%
5. 32-Pansini_Google #1 Resume Team _5_Night – 36%

None of the 3 candidates that that SME selected matched the Python Analyzer top 5.

4.3.3 Analysis of IBM Match Results

In Table 11 below, the numbers under the "PA Match" column show the match percentage of each resumes before synonyms are introduced. Numbers under the "PA Match with Synonyms" column show the match percentage of each resume after introducing matching based on keyword synonyms. The Blackbox ATS (BB ATS) column shows the average match percentage for all resumes using a BB ATS solution.

Table 11 Results from IBM Resume and Job Description MatchingTable 11

Resume Number	Black-Box ATS	PA Match	PA Match with Synonym	Change from Black-Box	Change from PA Match	% Change from PA Match
1-Mmandile IBM#1 Resume Team _1_Night	15.8	19	31	15.2	12	63.16%
2-Peck_IBM#1 Resume Team _1_Night	15.8	12	19	3.2	7	58.33%
3-Larsen_IBM#1 Resume Team _1_Night	15.8	27	38	22.2	11	40.74%
4-Podesk_IBM#1 Resume Team _2_Night	15.8	8	19	3.2	11	137.50%
5-Podesk_IBM#2 Resume Team _2_Night	15.8	4	15	-0.8	11	275.00%
6-Ebeling_IBM#1 Resume Team _3_Night	15.8	8	15	-0.8	7	87.50%
7-Neville_IBM#1 Resume Team _1_Day	15.8	8	19	3.2	11	137.50%
8-Mazut_IBM#1 Resume Team _1_Day	15.8	19	23	7.2	4	21.05%
9-Regan_IBM#1 Resume Team _2_Day	15.8	15	23	7.2	8	53.33%
10-Regan_IBM#2 Resume Team _2_Day	15.8	8	15	-0.8	7	87.50%
11-Regan_IBM#3 Resume Team _2_Day	15.8	15	19	3.2	4	26.67%
12-Regan_IBM#4 Resume Team _2_Day	15.8	4	12	-3.8	8	200.00%
13-Logue_IBM #1 Resume Team _3_Day	15.8	15	23	7.2	8	53.33%
14-Logue_IBM #2 Resume Team _3_Day	15.8	8	19	3.2	11	137.50%

15-Logue_IBM #3 Resume Team _3_Day	15.8	19	27	11.2	8	42.11%
16-Logue_IBM #4 Resume Team _3_Day	15.8	19	27	11.2	8	42.11%
17-Boley_IBM #1 Resume Team _4_Day	15.8	15	19	3.2	4	26.67%
18-Fren IBM #1 Resume Team _4_Day	15.8	12	15	-0.8	3	25.00%
19-Fren IBM #2 Resume Team _4_Day	15.8	8	23	7.2	15	187.50%
20-Kiss_IBM #1 Resume Team _4_Day	15.8	27	27	11.2	0	0.00%
21-Pere_IBM #1 Resume Team _4_Day	15.8	12	19	3.2	7	58.33%
22-Tariq_IBM #1 Resume Team _4_Day	15.8	23	31	15.2	8	34.78%
23-Pansini_IBM #1 Resume Team _5_Night	15.8	15	23	7.2	8	53.33%
24-DiChiara_ IBM#1 Resume Team _5_Night	15.8	15	23	7.2	8	53.33%
25-Herrera_IBM#1 Resume Team _4_Night	15.8	27	38	22.2	11	40.74%
26-Sumi_IBM#1 Resume Team _4_Night	15.8	27	38	22.2	11	40.74%
27-Attala_IBM#1 Resume Team _4_Night	15.8	23	31	15.2	8	34.78%
28-Ebeling_IBM#4 Resume Team _3_Night	15.8	19	31	15.2	12	63.16%
29-Ebeling_IBM#2 Resume Team _3_Night	15.8	8	23	7.2	15	187.50%
30-Ebeling_IBM#3 Resume Team _3_Night	15.8	15	23	7.2	8	53.33%
31-Viri_IBM#1 Resume Team _5_Night	15.8	15	27	11.2	12	80.00%
32-Alexis_IBM #1 Resume Team _5_Day	15.8	8	12	-3.8	4	50.00%

33-Derek_ IBM Resume#1_Team _5_Day	15.8	15	27	11.2	12	80.00%
34-Kati_IBM Resume_Team _5_Day	15.8	19	31	15.2	12	63.16%
35-Jacky_IBM resume_Team _5_Day	15.8	27	35	19.2	8	29.63%
Average	15.8	15.4	24.0	8.2	8.6	56.13

From a summary point of view, 35 IBM resumes were feed through the Python Analyzer (PA) and matches between the PA with keyword synonym matching and a BB ATS solution varied from -1% – 22.2% with an average change of a little more than 8%.

If we examine improvements as a percentage of the initial PA match on keywords alone, we see that algorithm adjustments to introduce matching based synonyms made a significant difference. In fact, at least 20% (7/35) of resumes showed improvements over 100% with the biggest improvement being 275% (from a 4% to 15% for 1 resume).

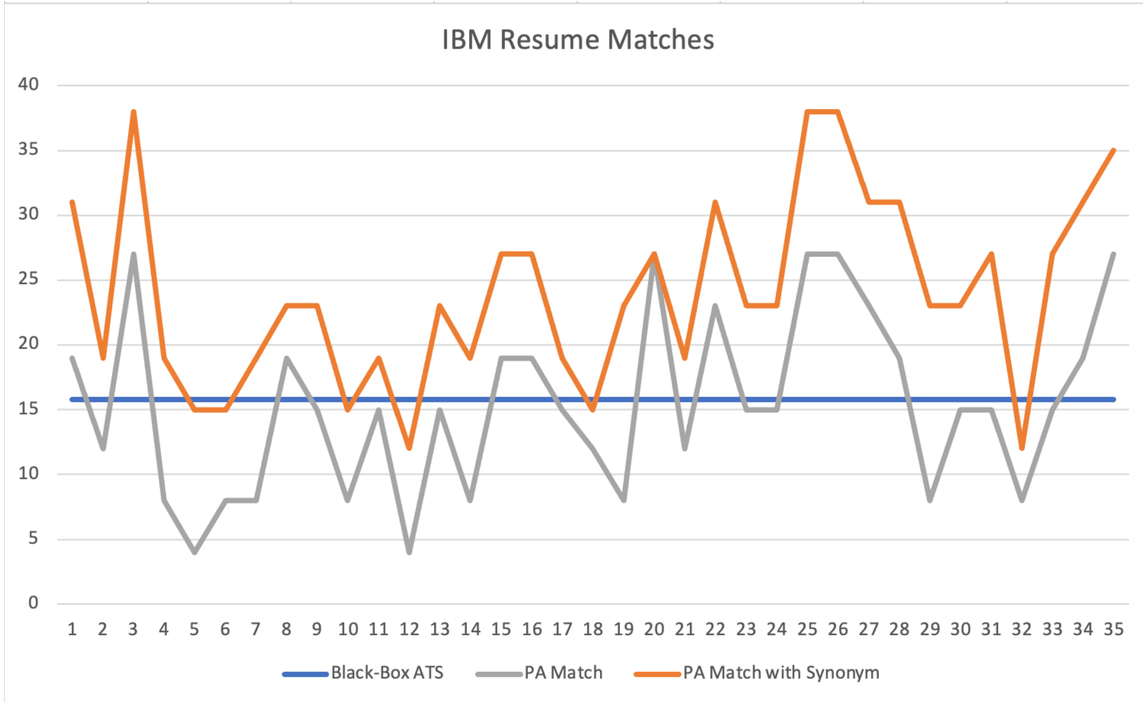

Figure 41 Line Graph of Resume Matches for IBMFigure 1

Regression Analysis

A review of Regression table below, indicates an R Square value of .789, which means that the predictor variable in the model (PA Match) explains 78.9% (.789x100) of the variance in the outcome variable PA Match With Synonyms.

Also note that the second column in the table contains coefficients and the fifth column contains p values associated with the regression. The coefficient for PA Match is .941 and has a p-value of .000 which is less than the cutoff (alpha level) of .05. The p-value means that this relation is statistically significant. Thus, the more that the matching algorithm makes use of synonyms, the more likely that this is an improvement in match percentage

Table 12 Regression Results from IBM Resume and Job Description MatchingTable 12

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	.888							
R Square	.789							
Adjusted R Square	.783							
Standard Error	3.387							
Observations	35.000							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	1.000	1415.489	1415.489	123.408	.000			
Residual	33.000	378.511	11.470					
Total	34.000	1794.000						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	9.543	1.422	6.712	.000	6.650	12.435	6.650	12.435
PA Match	.941	.085	11.109	.000	.768	1.113	.768	1.113

<u>Subject Matter Expert Analysis</u>

Part I:

I asked an IT professional who is AWS certified to review the 35 resumes for the IBM Cloud Software Developer job and list the top 3 - 5 candidates most qualified for the job. Below is a list of resumes that the IT professional provided:

Top 3:

1. Derek_ IBM Resume#1_Team _5_Day
2. Herrera_IBM#1 Resume Team _4_Night (25-Herrera_IBM#1 Resume Team _4_Night.docx)

3. Jacky_IBM resume_Team _5_Day (35-Jacky_IBM resume_Team _5_Day.docx)

The Python Analyzer's top 5 are listed below in order:

1. 26-Sumi_IBM#1 Resume Team _4_Night – 38%
2. 25-Herrera_IBM#1 Resume Team _4_Night – 38% - - → SME's #2
3. 3-Larsen_IBM#1 Resume Team_1_Night – 38%
4. 35-Jacky_IBM resume_Team _5_Day – 35% - - → SME's #3
5. 22-Tariq_IBM #1 Resume Team _4_Day – 31%

Two of the SME's candidates appears in the PA's top 5 list of candidates. This means that both sources thought that 86% of resumes were not the best match, and that of the remaining 14%, there is agreement on 40% (2 of 5) of candidates – see highlighted font above for details.

Part II:

I asked an IBMer with Cloud experience from the Office of the CIO to review the 35 resumes for the IBM Cloud Software Developer job and list the top 3 - 5 candidates most qualified for the job. Below is a list of resumes that the IBM professional provided:

Top 5:

1. Derek IBM #1 Derek_ IBM Resume#1_Team _5_Day

2. Sumi_IBM#1 Resume Team _4_Night ----> see below, there is a match (26-Sumi_IBM#1 Resume Team _4_Night)
3. Tariq_IBM #1 Resume Team _4_Day ---> see below, there is a match (22-Tariq_IBM #1 Resume Team _4_Day)
4. Attala_IBM#1 Resume Team _4_Night ---> see below, there is a match (27-Attala_IBM#1 Resume Team _4_Night)
5. Ebeling_IBM#3 Resume Team _3_Night

The Python Analyzer's top 5 are listed below in order:

1. 26-Sumi_IBM#1 Resume Team _4_Night – 38% -- → SME's #2
2. 25-Herrera_IBM#1 Resume Team _4_Night – 38%
3. 3-Larsen_IBM#1 Resume Team_1_Night – 38%
4. 35-Jacky_IBM resume_Team _5_Day – 35%
5. 22-Tariq_IBM #1 Resume Team _4_Day – 32% --→ SME's #3

Two of the SME's candidates appears in the PA's top 5 list of candidates. This means that both sources thought that 86% of resumes were not the best match, and that of the remaining 14%, there is agreement on 40% (2 of 5) of candidates – see highlighted font above for details.

Note: The IBMer indicates that his initial list had all 4 of the 5 names in the PA list.

Summary:

This chapter describes results from approximately 120 resumes being processed by the the Python Analyzer (RA). A table captures results for each resume processed for job descriptions from Amazon, Google and IBM. The table also shows match percentages before and after synonyms are used for matching of resumes against job description. Summary statistic describing improvement in match percentages are listed. A regression analysis also shows that match improvements observed is not due to chance. Finally, I asked Subject Matter Experts (SMEs) to review resumes again job description and identify their top 3 to 5 top candidates. Results show that the SMEs and PA tool both eliminated over 80% of the same candidates, and matched on more than 40% of the top 5 candidates.

Chapter 5 Conclusion

5.1 Achievements

This research has shown that the quality of job application preprocessing can be improved when keywords and their synonyms derived from Knowledge are used when seeking a match between resumes and a job description. Match percentage for approximately 120 resumes were reviewed before and after adjusting an ATS program called Python Analyzer (PA) to account for Knowledge Graph keywords and keyword synonyms. When Knowledge Graph keywords and synonyms were applied to the 3 job descriptions from Amazon, Google and IBM, results showed improvement for almost every resume processed. Regression analysis for each job description also showed that the observed differences in match percentage were not due to chance (p-values were less than .05). Further, Subject Matter Experts (SMEs) who are practitioners in the cloud space selected many of the same candidates as the PA for their top 3 to 5 best candidates. Both SMEs and the PA eliminated the same 85% of candidates and selected the same top 15% of resumes 30 to 60% of the time. Since SMEs had no knowledge of PA's top candidates, output from both sources corroborated each other. This is evidence that results from the PA ATS is credible.

Given results described above, there is substantial evident that the quality of job application pre-processing can be improved with keywords and synonyms derived from Knowledge Graphs. The PA ATS saves time by automating the resume filtering process thereby allowing HR personnel to sort through more resumes faster. Research results are conclusive that the use of Knowledge Graph keywords and synonyms during matching helped to identify the best list of candidates for the 3 job descriptions in the experiment.

In addition to helping the HR team same time, the PA framework which includes the Knowledge Graph visual also helps HR personnel by highlighting key technical relationships that may not be obvious to non-technical personnel who is not trained in a software discipline.

5.2 Future Work

There are a number of areas that may be good for future work. Below is a list of opportunities for future work:

- **Standardizing resume formats for pre-processing** – Resumes are written in many formats (i.e. .doc, .pdf, .page, etc.) and there may be a reduction in format processing errors if all resumes have the same or a similar format prior to starting the matching process analysis.
- **Post hire analysis** – Since some hires do not work out, analyzing failure rate for those screened by the PA ATS versus other filtering sources may be good to investigate.
- **ATS integration with Legacy Systems** – Given the complexity of ATS and the increasing number of new features, investigating integration challenges associated with ATS software could be good follow-on work.

Appendix A Tariq_Amazon #1 Resume Team _4_Day

EDUCATION
Marist College, Poughkeepsie, NY
Bachelor of Science in Computer Science, Emphasis in software development
GPA: 3.8, (May 2021)

HONORS AND AWARDS
Presidential Scholarship

RELEVANT EXPERIENCE
Internship Project, DockYard
Custom design for IT startup,

- Provided exceptional professional services in strategy, user experience and design
- Built a SaaS platform that can manage e-commerce, POS, and data analytics.

LEADERSHIP **Other Activities**

- Muslim Student Association, Vice President (2019)
- Feed the World, non-profit social program, Head Teacher for female students (2016-2017)
- WWF-Pakistan's International Eco Internship Programme (2016)

WORK EXPERIENCE
ELEKS, Las Vegas, NV
Software Developer, (August 2015- 2019)

- Provided clients with software solutions for unparalleled business growth
- Expert on a wide array of technologies, including .NET, front end, Java, databases, big data, mobile and DevOps.

Marist College Housing Office, Poughkeepsie, New York
Resident Assistant, (2014-2015)

- Acted as a resource, mentor, and community leader in residence area.
- Served a diverse student population, and clearly demonstrate an understanding of multiculturalism.

Marist College IT Help Desk, Poughkeepsie, NY
IT Helper, (May 20th-July 1st 2014)

- Addressed technical issues such as account services, client computing services, networking services to assist students.
- Attended IT training to enhance my skills as an IT Helper.

Marist Poll, Poughkeepsie, NY
Interviewer, (Nov 13th,2013-May 17,2014)
Served as an interviewer on a team of 300 to solicit public opinions for research.

SKILLS
Strong working knowledge of Microsoft Office: Excel, PowerPoint, Word
Java/J2EE (expert), HTML/CSS (expert), AngularJS (moderate), SQL (beginner)

LANGUAGES
Fluent in Urdu, Punjabi and Hindi; Intermediate Proficiency in Arabic

Appendix B

Ebeling_Google#2 Resume Team _3_Night

EDUCATION

Cloud College, Poughkeepsie, NY
Bachelor of Science in Computer Science, Concentration in Software Systems
May 2019, GPA: 3.83

HONORS AND AWARDS

- Dean's List, Spring 2017, Fall/Spring 2018, Spring 2019
- National Merit Award

RELEVANT EXPERIENCE

Comcast Tech Software Engineer Internship
Software Intern, May 2018-August 2018

- Maintained code base of creative services and transcoding components
- Developed unit and functional tests against cases and maintained a regression test framework
- Gained foundation in data structures, algorithms, and software design
- Developed cloud-based web application server prototypes and applications for mobile and internet cloud technologies
- Participated in projects involving writing, debugging, and optimizing codes.

Interactive Brokers Group, Inc. Software Engineer Summer Internship
Consulting Intern, June 2017-August 2017

- Developed high-performance, large data research platforms
- Participated and collaborated in teams to generate ideas regarding large scale system design and user interface design.
- Gained an understanding of data structures and algorithms of applications and
- Built, designed and implemented scalable cloud-based web applications.

LEADERSHIP ACTIVITIES

National Computing Organization
Member August 2016 - May 2019
Vice President, January 2018 - May 2019

- Planned, organized and led development activities for the organization.
- Communicated and networked with students, faculty, and alumni in the organization.
- Networked with faculty and professionals within the computer science industry.
- Assisted organization members with issues concerning academics, personal or professional.

Best Buddies
Club Member, January 2016 - May 2019

- Attended board meetings and aided in organizing, planning, and setting goals.
- Volunteered and participated in campus and community events held by the club.

- Interacted and assisted with special needs kids and adults in the community.

WORK EXPERIENCE

Cloud College IT Help Desk, Poughkeepsie, NY
IT Helper, September 2016-May 2019

- Address technical issues and provide resolutions and reports of issues found.
- Assisted students, faculty and staff with their questions, request, and suggestions.

SKILLS

- Competent with the use of Java, C/C++, Python, Go, database design, HTML, ASP, and SQL.
- Knowledge experience of TCP/IP, Unix/Linux, and network programming.
- Ability to write computer programs in Python and VBA.
- Proficient in Microsoft Office (Excel, Word, Access, PowerPoint) and Adobe Creative Cloud (Photoshop, Illustrator, InDesign, Premiere Pro)
- Capability to problem solve, prioritize, communicate, multi-task, and manage resources
- Ability to identify problems, gather information, develop evaluation, and calculate results.
- Familiar with open source cloud and application platforms (Google App Engine)

Appendix C

Jacky_IBM resume_Team _5_Day

EDUCATION

Cloud College, Poughkeepsie, NY
Bachelor Degree of Computer Science, Emphasis in Computer Engineering
May 2018, GPA: 3.7

HONORS AND AWARDS

- Dean's List, Spring 2018
- New York Society of CPAs College Scholarship, April 2016
- Distinguished Scholars List, May 2018

RELEVANT EXPERIENCE

Google, LLC Summer Internship Program
Consulting Intern, June 2017-August 2017

- Wrote code for Google search engine
- Utilized all kinds of programing languages such as PHP, Python, and JavaScript.

IBM Programming Program
Software Developer, June 2017-August 2017

- Wrote code for IBM Cloud
- Utilized advanced administration and automation of virtualization such as KVM and Xen.

Amazon Consulting, LLC Winter Externship Program
Extern, January 2017

- Participated in AWS rotation program to learn EC2

LEADERSHIP ACTIVITIES

National Computing Honor Organization
Club Member September 2014 – May 2018
National Member May 2017 - present
President, January 2017-Present

- Plan and organize professional development activities for the organization.
- Communicate with students, officers, academic advisors, and recruiting professionals daily.
- Network and learn from professionals within the computer science industry.
- Tutor academically struggling accounting students every week.

Greek Honor Society
Member, January 2018 - present

- Attended board meetings to aid in organizing, planning, and setting goals.
- Assisted in prioritizing and coordinating various campus and community events.

WORK EXPERIENCE

College IT Help Desk, Poughkeepsie, NY
IT Helper, September 2015-Present

- Address technical issues to assist students in resolving technology issues with their computer
- Attended IT training to enhance my skills as an IT Helper.

SKILLS

- Can write computer programs in Python, and VBA
- Strong working knowledge of Microsoft Office (Excel, Word, Access, PowerPoint).
- Ability to prioritize, multi-task, and manage resources effectively and efficiently.
- Familiar with both Windows and Linux system administrations.
- Familiar with programing languages such as PHP, Python, C#, JavaScript, MariaDB, and MongoD

Appendix D Python Code

```
import os
import docx2txt
import glob
import pandas as pd

import string

from datamuse import datamuse

api = datamuse.Datamuse()

# Indicate if processing for IBM, Amazon or Google
userChoice = input("enter 1 for IBM, 2 for Amazon, 3 for Google :").strip()
# print(userChoice)

# Processing continues for IBM, Amazon or Google nest all
if userChoice in set(['1', '2', '3']):
    print("... loading data please wait... (should take ~5 minutes) ")
    # inputVariable is set to search directory data1, etc
    inputVariable = 'data' + userChoice

    # Structures for KG and Job Description keyword sets
    KGraph_kw_Set = JobDesc_kw_Set = set()

    # use drop to clean data, drop commas, periods
    drop = string.punctuation + string.digits

    # i create a new set to put all the keywords in
    Matching_KG_JobDes_kw = set()
    nomatchkeywords = []

    keywords_synonym = set()
    # populate "KGraph_kw-set" with external text file
    KGraph_kw = open('keywords.txt').read().lower().split()
    KGraph_kw_Set = set(word.strip(drop) for word in KGraph_kw)
    # populate "JobDes_kw_set"
    JobDesc_kw = open(inputVariable + '.txt').read().lower().split()
    JobDesc_kw_Set = set(word.strip(drop) for word in JobDesc_kw)

    # check for similarity( aka if the word in question,  exists in the other set

    # loop over each word in KGraph_kw_set
    for word in KGraph_kw_Set:
        # if word from KGraph_kw_set is in JobDes_kw_set
        if word in JobDesc_kw_Set:
            # place it in Matching_KG_JobDes_kw
            Matching_KG_JobDes_kw.add(word)
```

```
    # print(Matching_KG_JobDes_kw)

    # read in each resume and convert it to text

# path may need to change – see comment below

    os.chdir("/Users/josephporter/PycharmProjects/test/Data/" + inputVariable)

    # InputVariable = data1 for IBM, etc.

    # Create structures for resumes converted to text #
    Descriptions = []
    name = []
    # synPerc = 0;

    for file in glob.glob('*.docx'):
        Descriptions.append(docx2txt.process(file))
        name.append(file)

    data = pd.DataFrame(
        {'Descriptions': Descriptions,
         'Name': name,

         })
    print("heavy processing, feel free to grab a coffee")

    MatchCount_KG_JobDes_kw = len(Matching_KG_JobDes_kw)

    # print(len(Matching_KG_JobDes_kw))

    # create structure for totals
    MatchCount_KG_JobDes_Res_kw_Total = []
    MatchCount_Syn_KG_JobDes_Res_kw_Total = []

    # loop through resumes
    for i in range(len(data)):
        MatchCount_KG_JobDes_Res_kw = set()

        # populate Resume_kw_set from prior structure
        Resume_kw_set = data.loc[i]['Descriptions']

        for word in Matching_KG_JobDes_kw:
            if word in Resume_kw_set:
                MatchCount_KG_JobDes_Res_kw.add(word)
        MatchCount_KG_JobDes_Res_kw_Total.append(MatchCount_KG_JobDes_Res_kw)

        # create structure
        UnMatch_KG_JobDes_Res_kw = []
        UnMatch_Syn_KG_JobDes_Res_kw = []
```

```
        for word in Matching_KG_JobDes_kw:
            if word not in MatchCount_KG_JobDes_Res_kw:
                UnMatch_KG_JobDes_Res_kw.append(word)

        # print(len(UnMatch_KG_JobDes_Res_kw))
        for word in UnMatch_KG_JobDes_Res_kw:
            UnMatch_Syn_KG_JobDes_Res_kw.append(api.words(rel_syn=word, max=5))
        # structure to hold 5 synonyms for unmatch words
        unmatchedSynonymMasterList = pd.DataFrame({
            'word': UnMatch_KG_JobDes_Res_kw,
            'SynonymList': UnMatch_Syn_KG_JobDes_Res_kw
        })

        synonymHits = []
        print('... Still processing, Hang in there ... \n')
        for f in range(len(unmatchedSynonymMasterList)):
            for synonym in unmatchedSynonymMasterList.loc[f]['SynonymList']:
                if synonym['word'] in Resume_kw_set:
                    if unmatchedSynonymMasterList.loc[f]['word'] not in synonymHits:
                        synonymHits.append(unmatchedSynonymMasterList.loc[f]['word'])
        # Update ######################################################
        # print(synonymHits)
        # prints out list of words found with
Synonyms######################################

        for word in MatchCount_KG_JobDes_Res_kw:
            synonymHits.append(word)
        MatchCount_Syn_KG_JobDes_Res_kw_Total.append(synonymHits)

    # print(MatchCount_KG_JobDes_Res_kw_Total)
    # print(MatchCount_Syn_KG_JobDes_Res_kw_Total)
    SynPerc =[]
    for k in range(len(MatchCount_KG_JobDes_Res_kw_Total)):
        matchPercentage = 0
        matchPercentage = len(MatchCount_KG_JobDes_Res_kw_Total[k]) /
MatchCount_KG_JobDes_kw
        print("Match Percentage: %.2f from document %s" % (matchPercentage * 100,
data.loc[k]["Name"]))
        synMatchPercentage = 0
        synMatchPercentage = len(MatchCount_Syn_KG_JobDes_Res_kw_Total[k]) /
MatchCount_KG_JobDes_kw
       #should update data with synonym match score under 'syn_perc' column name
        SynPerc.append(synMatchPercentage *100)
        print("Match Percentage with Synonyms: %.2f from document %s \n" %
(synMatchPercentage * 100, data.loc[k]["Name"]))
    data2 = pd.DataFrame(
        {'Name': name,
         'Syn_Perc': SynPerc
         })
    data2 = data2.nlargest(10, 'Syn_Perc')
```

```
    print('Top 10 Candidates:')
    # for v in range(10):
    #     print("Match Percentage with Synonyms: %.2f from document %s \n" %
(data2['Syn_Perc'] * 100, data2["Name"]))
    print(data2)

else:
    print("you have entered an invalid choice")
```

References

[1] Ahmed, N., Khan, S., Latif, K., "Job Description Ontology," Communications from 2016 International Conference on Frontiers of Information Technology, [Online]. Page 5 of article, page 221 from consolidated materials

[2] Augustine, A.C., "What Is an ATS? How to Write a Resume to Beat the Bots,"December 22, 2017, [Online]. Available: https://www.topresume.com/career-advice/what-is-an-ats-resume [Accessed February 2020]

[3] Barcelos, K., "Top 6 Technology Skills Every HR Professional Needs Today," Available: https://www.digitalhrtech.com/technology-skills-every-hr-professional-needs/

[4] Bell, R., "Artificial Intelligence, Automation and the Future of Talent Acquisition(Technology)," June 15, 2018. [Online]. Available: https://www.workforce.com/2018/06/15/artificial-intelligence-automation-and-the-future-of-talent-acquisition/ . [Accessed October 2018].

[5] Bersin, J., "HR Technology for 2018: Ten Disruptions Ahead," November 2, 2017 [Online]. Available: https://www.forbes.com/sites/joshbersin/2017/11/02/hr-technology-for-2018-ten-disruptions-ahead/#6e4ea14443b7 [Accessed March 2020]

[6] Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., Eckstein, R., "The Impact of Semantic Web Technologies on Job Recruitment Processes," Available: http://page.mi.fu-berlin.de/mochol/papers/wi2005.pdf

[7] Chang, L., "Applying data mining to predict college admissions yield: A case study," 2006. New Dir. Institutional Res., 2006 (131): 53–68 Available: https://ieeexplore-ieee-org.rlib.pace.edu/stamp/stamp.jsp?tp=&arnumber=7429390

[8] Costa, D., "How Artificial Intelligence improves your recruitment process," December 22, 2017,[Online]. Available: https://www.skeeled.com/blog/how-artificial-intelligence-improves-your-recruitment-process/ [Accessed February 2020]

[9] Dadiani, E., "Job Ontology – Key to Success in Recruiting Domain," March 07, 2018. [Online]. Available: https://www.jobontology.com/job-ontology-key-to-success-in-recruiting-domain-3/ . [Accessed November 2019]

[10] Devanbu, P., Brachman, R.J., Selfridge, P.G., and Ballard, LASSIE, B.W.B,: "A knowledge-based software information system." Communications of the ACM, 34(5):34-49, 199

[11] ERE, "3 Keys to recruiting women for your workforce," 2017, Available: www.eremedia.com

[12] Gaucher, D., Friesen, J., & Kay, A. C., "Evidence that gendered wording in job advertisements exist and sustains gender inequality. Journal of Personality and Social Psychology, 101, 109-128. 2011; Wild, J., "Wanted-a way with words in recruiting

ads," March 7, 2017. Available: Financial Times. www.ft.com/content/9974b0ce-e7bb-11e6-967b-c88452263daf

[13] Gould, S., "Agile Co-Location: Why It Works," January 10, 2017 Available: https://www.linkedin.com/pulse/agile-co-location-why-works-scott-gould

[14] Hiers, M., "What 'Job Matching' Really Means," 2014 [Online]. Available: https://www.pandologic.com/job-boards/job-matching-really-means/ [Accessed March 2020]

[15] Holzinger, A., Simonic, K., Yildirim, P., "Disease-disease relationships for rheumatic diseases - Web-based biomedical textmining and knowledge discovery to assist medical decision making," 2012. Available: https://ieeexplore-ieee-org.rlib.pace.edu/search/searchresult.jsp?newsearch=true&queryText=medical%20disease%20keyword&highlight=true&returnType=SEARCH&matchPubs=true&pageNumber=1&returnFacets=ALL

[16] Hunter, A., " Intelligent Text Handling Using Default Logic," IEEE Xplore

[17] Hunter, A., " Uncertainty in Information Systems. McGraw-, and Parallel/Distributed Computing," 1996. McGraw-Hill

[18] Kelley, A., "CareerBuilder Deepens Investment in Talent Acquisition R&D and Industry-Leading Artificial Intelligence Capabilities," January 23, 2019. [Online]. Available: http://press.careerbuilder.com/2019-01-23-CareerBuilder-Deepens-Investment-in-Talent-Acquisition-R-D-and-Industry-Leading-Artificial-Intelligence-Capabilities . [Accessed November 2019]

[19] MacGillivray, E.D., Beecher, J. H., Golden, D.M, "Employment Testing: The New Hot Button Issues for Federal Agencies and - Other Legal Departments, " March – April 2008. Global Business & Organizational Excellence 27, no. 3, pages 68 – 78

[20] Mochol, M., Simperl, E.P.B, "P"Practical Guidelines for Building Semantic eRecruitment Applications," Available https://www.researchgate.net/profile/Elena_Simperl/publication/228610902_Practical_Guidelines_for_Building_Semantic_eRecruitment_Applications/links/572b964608ae2efbfdbdd9f3.pdf

[21] Nonaka, I., Takeuchi, H., "The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation," 62

[22] Ongig team, R. "The Top 100 Applicant Tracking Systems in 2018," [Online]. Available: https://blog.ongig.com/recruiting-software/top-100-applicant-tracking-systems-in-2018/ [Accessed February 2020]

[23] Pearlson, K., Saunders, C., "Managing & Using Information Systems," 2013. Pages 332 - 334

[24] Quantz, J., Schmitz, B., "Knowledge-based disambiguation for machine translation. Minds and Machines," 4:39-57,1994: Link? . [Accessed]

[25] Sakolnakorn, P.P.N.,Meesad, P.,“ Decision Tree-Based Model for Automatic Assignment of IT Service Desk Outsourcing in Banking Business,” Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

[26] Sette,M.,"2017-1-9 MARIA_SETTE_DISSERTATION_FINAL_DRAFT_JAN_9_2017_V11 1215AM, page 50 - 57 “ January 09, 2017. [Online]. Available: . [Accessed]

[27] SHRM, “Talent acquisition: Selection. HRToday,” 2016 www.shrm.org/hr-today/trends-and-forecasting/research-and-surveys/Documents/Talent-Acquisiton-Selection.pdf

[28] Snell, S., Morris, S., Bohlander. G., “Managing Human Resources” 2014, 17^{th} edition, page 224

[29] Sullivan, J. “Recruiting has the highest business impact of any HR function,” 2012 [Online].Available:www.bcg.com/expertise/capabilities/people-organization/human-resources.aspx

[30] The Martec Group, “ 2017 Recruiter Sentiment Study,” 2017 [Online]. Available: https://www.mrinetwork.com/media/304094/2017hiringsentimentstudy.pdf , Page 7 of 21 [Accessed March 2020]

[31] Urbano, J., “How AI-based automation of the matching process improves recruitment,” July 3, 2018 [Online]. Available: https://www.skeeled.com/blog/how-ai-based-automation-of-the-matching-process-improves-recruitment/ [Accessed March 2020]

[32] West, M., “People Analytics for Dummies,” 2019. Page 23

[33] Won, H., Lee, Y., Shim, J.,Ahn, H., “A Hybrid Collaborative Filtering Model Using Customer Search Keyword Data for Product Recommendation,” 2019. 18th IEEE International Conference on Machine Learning and Applications (ICMLA)

[34] Wright, J.R., Weixelbaum, E.S., Brown, K., Vesonder, G.T., Palmer, S.R., Berman, J.T., Moore, H.H., “A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems.” AI Magazine, 14(3):69-80, 1993