2018 IEEE International Conference on Data Mining Workshops (ICDMW)

Generating Unified Candidate Skill Graph for Career Path Recommendation

1st Akshay Gugnani
IBM Research-India
Bangalore, India
akgugnan@in.ibm.com

2nd Vinay Kumar Reddy Kasireddy
IBM T J Watson Research Centre, YorkTown Heights
NY, USA
vinay.kasireddy@ibm.com

3rd Karthikeyan Ponnalagu
Bosch India
Bangalore, India
karthikeyan.ponnalagu@gmail.com

***Abstract*—"How should I progress in my career?" is an important question that every working professional seeks answer multiple times during her career. Given the amount of career position data of individuals available online, personalized career path recommendation systems that could mine and recommend the most relevant career paths for a user are on the rise. However, such recommendation systems typically are only effective within a single organization where there are standardized job roles. At an industry sector level such as Information Technology or across such different industry sectors (such as retail, insurance, health care), mining and recommending the most relevant career paths for a user is still an unsolved research challenge. Towards addressing this problem, we propose a system that leverages the notion of skills to construct skill graphs that can form the basis for career path recommendations. We perceive skills are more amenable for career path standardizations across the organizations. Our proposed system ingests a users profile (in a pdf, word format or other public and shared data sources) and leverages an Open IE pipeline to extract education and experiences. Subsequently, the extracted entities are mapped as specific skills that are expressed in the form of a novel unified skill graph. We believe that such skill graphs which capture both spatial and temporal relationships aid in generating precise career path recommendations. An evaluation of our current skill extraction model with an industrial scale dataset yielded a precision and recall of 80.54% and 86.44% respectively.**

***Index Terms*—Skill graph, Text to Graph extraction, Career path recommendation systems, NLP, Data mining**

I. Introduction

Today a significant population of working professionals publish their career position data in social media-centric professional networks. As each and every individual is unique in terms of skills, experience and education, personalized career path recommendation systems that could mine such available career position data and recommend the most relevant career paths for a user are becoming popular. However, these recommendation systems usually are effective within a single organizational context where the notion of career paths and job roles are well defined but not at an industrial sector level across organizations [14]. This is primarily due to the lack of standardized definitions of jobs and career roles across organizations. For example: a *software engineer* role in one organization might be performing job duties similar to that of a *system analyst* role in a different organization. Therefore movements between such semantically similar roles cannot be considered as part of career progression related recommendations. To address this problem, we propose to formalize skills as the building blocks for career move recommendations as they are more amenable to standardization across organizations. We extract each and every information (whether it is education or experience related) from the user profile and generate a skill graph consisting of similarity, parent-child and dependency type relationships. To infer such relationships, we have built a skill ontology using a list of standardized data sources available online. Each node in the skill graph is also annotated with the expertise level information (how proficient the user is with respect to a skill). Career path recommendation system built on such enriched skill graphs could help in identifying the skill based transitions that user could consider and the associated job roles or positions for such transitions. Since such recommendations are skill based, we believe these would be more useful and easy to follow as actionable items for career advancement.
The rest of the paper is organized as follows: Section 2 presents the related work, section 3 discusses the technical details of the system and section 4 presents the future work and conclusions.

II. Related Work

Field of automated career path recommendation systems is relatively unexplored. Verma et al. [14] propose a system to recommend personalized career paths using similarities across education and experience. Such work assumes availability of standardized notion of job roles. Razak et al. [13] leverages user survey data of personality and skills to build a fuzzy logic model for generating career path recommendations. Skill extraction as a topic however has garnered enough attention. Kivimaki et al. [7] propose a system for skill extraction from documents primarily targeting towards hiring and capacity management in an organization. The work initially computes similarities between an input document and the texts of Wikipedia pages and then uses a biased, hub-avoiding version of the Spreading Activation algorithm on the Wikipedia graph to associate the input document with skills. Wang et al. [15] propose a system to extract skills from user profiles in a social network context. First, personal connections with similar academic and business background (e.g. co-major, co-university, and co-corporation) are extracted and then the skill connections between skills from the same person. Faizan et

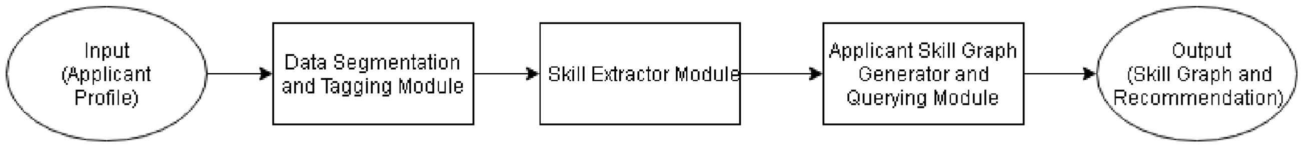


Fig. 1. Proposed System Overview

al. [6] use a named entity normalization system for detecting occupational skills in resumes or job ads. Their method employs a skills tagger that uses properties of semantic word vectors to recognize and normalize relevant skills, and then a skill entity sense disambiguation component to infer the correct meaning of an identified skill.

Skill management systems that employ skill ontologies are also popular among organizations [2]. Fotis et al. [3] present an ontology based application for competency management and learning paths. Lau et al. [9] present a methodology for application-driven development of ontologies, with a sample instantiation of the methodology for skills ontology development. These skill management systems although help in specifying skills at various proficiency levels, they do not automatically infer skills. It is up to users to provide their skills data to the system as per the defined ontology. There is also a lot of work done on job recommender systems [5] [10] [12] [8] [1] which match job descriptions with that of user profiles. However, none of them explore the idea of building a skill graph automatically from career position data to address the problem of career path recommendations.

III. System Overview

This section discusses the architecture of the proposed formal machinery for constructing skill graphs. Figure 2 depicts the architecture of the system. The input to the proposed system is user profile documents. The user profile could be in the form of pdf or word doc formats or text accessible from social media profile URLs. At a high level, our system consists of three major modules:

- Data Segmentation and Tagging Module
- Skill Extraction Module
- Skill Graph Generator Module

Figure 1 shows the overall system overview. Each of these modules is elaborated in the following subsections.

A. Data Segmentation and Tagging Module

Based on the input selected, if the applicant profile is a document, the system uploads the document to Watson Discovery Service [1]. The IBM Watson Discovery service converts a single HTML, PDF, or Microsoft Word document into normalized HTML, plain text, or a set of JSON-formatted Answer Units. We use this to convert the different input document formats to a more standardized output text. If the input is an online reference (on publicly available sites), the link is passed to a custom web scraping API, which extracts and stores all relevant content from the website in a text file. Regardless of the input format, at this stage, the system has the input converted in the form of a unified text representation.

The text representation is then passed to "Data Segmentation and Tagging Module". For segmentation and tagging, we manually evaluated hundreds of resumes and based on common practices in resume writing, we generalized and defined a set of heuristics to identify the following segments and their details.

- **Candidate Details:** Name, Email, Contact info, Websites, DoB
- **Education and Academic Details:** Institute, Degree, GPA, courses, degree duration, publication/patent
- **Experience:** Job/Role (such as internship, position held, volunteer work), duration, Organization

We assume between every consecutive set of dates there is a unique segment of the user profile. Each position present in the profile is treated as a candidate item for professional improvement inference. Example: current position and previous positions (responsibilities and projects done as part of the positions) each is evaluated and measured. Figure 3 depicts the various tagged segments of the candidate resume.

The first segment of the resume, usually about the first 15-20 words has the candidate personal details. To identify and distinguish the candidate's name from other proper nouns, we use Stanford core NLP [11] to identify noun phrase within the first 20 words. The first occurrence of noun phrase validated by Watson Natural Language Understanding Service[2] as a name-type entity is established as the candidate name. We use Watson NLU's Named Entity Recognizer to validate the noun phrases. In order to identify the email id, contact number, website and date of birth, we use an exhaustive case of regular expressions(regex) to identify their occurrence within the document. In Figure 3, the purple index highlights the candidate details that would be extracted.

In order to classify a segment as Education and Academic detail, we parse the segment of text to check for the presence of institute/university name, GPA, degree name and/or courses. If all the above co-occur, the unit is classified as a Education and Academic detail segment. For example, the education assessment scores such as The GPA/CGPA is identified using regex and/or occurrence of the term GPA /CGPA or other form of grade/scores like the SAT/ACT. We validate institute/university name and degree/course name through our knowledge base

[1]https://www.ibm.com/watson/services/discovery/

[2]https://natural-language-understanding-demo.ng.bluemix.net/

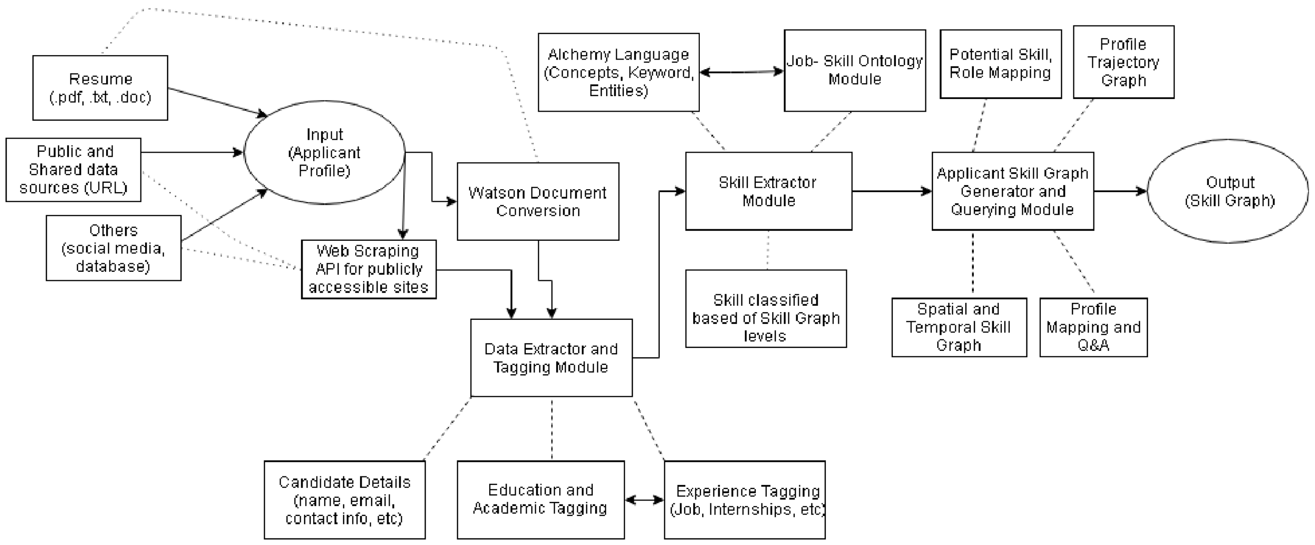
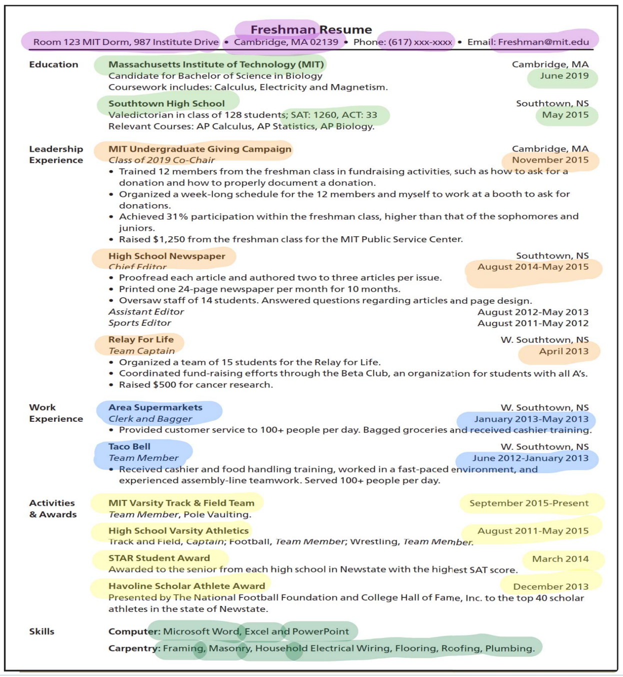

Resume (.pdf, .txt, .doc)
Public and Shared data sources (URL)
Others (social media, database)
Input (Applicant Profile)
Web Scraping API for publicly accessible sites
Watson Document Conversion
Data Extractor and Tagging Module
Candidate Details (name, email, contact info, etc)
Education and Academic Tagging
Experience Tagging (Job, Internships, etc)
Alchemy Language (Concepts, Keyword, Entities)
Job- Skill Ontology Module
Skill Extractor Module
Skill classified based of Skill Graph levels
Potential Skill, Role Mapping
Profile Trajectory Graph
Applicant Skill Graph Generator and Querying Module
Spatial and Temporal Skill Graph
Profile Mapping and Q&A
Output (Skill Graph)

Fig. 2. System Architecture and Flow Diagram

and Watson Natural Language Understanding service API. We employ a feedback mechanism to learn and update new data as identified by Watson Natural Language Understanding service API. As depicted in Figure 3, the light-green index highlights the experience classified as two units of Education i.e. his university and school.



Freshman Resume
Room 123 MIT Dorm, 987 Institute Drive • Cambridge, MA 02139 • Phone: (617) xxx-xxxx • Email: Freshman@mit.edu

Education

Massachusetts Institute of Technology (MIT) Cambridge, MA
Candidate for Bachelor of Science in Biology June 2019
Coursework includes: Calculus, Electricity and Magnetism.

Southtown High School Southtown, NS
Valedictorian in class of 128 students; SAT: 1260, ACT: 33 May 2015
Relevant Courses: AP Calculus, AP Statistics, AP Biology.

Leadership Experience

MIT Undergraduate Giving Campaign Cambridge, MA
Class of 2019 Co-Chair November 2015
- Trained 12 members from the freshman class in fundraising activities, such as how to ask for a donation and how to properly document a donation.
- Organized a week-long schedule for the 12 members and myself to work at a booth to ask for donations.
- Achieved 31% participation within the freshman class, higher than that of the sophomores and juniors.
- Raised $1,250 from the freshman class for the MIT Public Service Center.

High School Newspaper Southtown, NS
Chief Editor August 2014-May 2015
- Proofread each article and authored two to three articles per issue.
- Printed one 24-page newspaper per month for 10 months.
- Oversaw staff of 14 students. Answered questions regarding articles and page design.

Assistant Editor August 2012-May 2013
Sports Editor August 2011-May 2012

Relay For Life W. Southtown, NS
Team Captain April 2013
- Organized a team of 15 students for the Relay for Life.
- Coordinated fund-raising efforts through the Beta Club, an organization for students with all A's.
- Raised $500 for cancer research.

Work Experience

Area Supermarkets W. Southtown, NS
Clerk and Bagger January 2013-May 2013
- Provided customer service to 100+ people per day. Bagged groceries and received cashier training.

Taco Bell W. Southtown, NS
Team Member June 2012-January 2013
- Received cashier and food handling training, worked in a fast-paced environment, and experienced assembly-line teamwork. Served 100+ people per day.

Activities & Awards

MIT Varsity Track & Field Team September 2015-Present
Team Member, Pole Vaulting.

High School Varsity Athletics August 2011-May 2015
Track and Field, *Captain*; Football, *Team Member*; Wrestling, *Team Member*.

STAR Student Award March 2014
Awarded to the senior from each high school in Newstate with the highest SAT score.

Havoline Scholar Athlete Award December 2013
Presented by The National Football Foundation and College Hall of Fame, Inc. to the top 40 scholar athletes in the state of Newstate.

Skills

Computer: Microsoft Word, Excel and PowerPoint
Carpentry: Framing, Masonry, Household Electrical Wiring, Flooring, Roofing, Plumbing.

Fig. 3. Segmentation and Tagging on a Sample Resume

A segment of data which does not start with an institute name or degree name or the candidate details is identified as Experience segment. The segment is further sub-classified as Internship if the date of this segment lies within the dates of the Education segment. It is classified as job role if there is a mention of a company/industry name which is identified by Named Entity Recognizer(NER) and our existing knowledge database. It is classified as a publication based on the occurrence of an unique identifier such as ISBN, keyword match to conference/publication presence of a URL.

The entity and keyword details such as nouns are validated through Watson Natural Language Understanding service and the feedback system keeps enhancing the knowledge database as more profiles are parsed. Each segment that is appropriately tagged is then passed to the skill extraction module. It is interesting to note, In Figure 3, the content in dark-green index (bottom of the figure) does not fall in any predefined rules but is rich with skills. In such a case, these are added as overall potential skills of the candidate independently and not mapped to a specific category.

The data segmentation and tagging module is a crucial step as it forms the temporal structure of our skill graph. Most application tracking systems consider a candidate resume as a single entity and extract keyword and skill which are labeled to the profile, while we leverage the segmented profile data to form the skill graph. It is important to correctly identify and tag candidate profile details such as their university, courses taken, various experiences and their personal details to correctly identify the skills that were acquired at each stage and leverage this skill sequence information with additional heuristics as a key distinguisher to make better skill recommendations.

B. Skill Extraction Module

This section discusses the techniques developed for extracting skills from a segmented candidate profile.

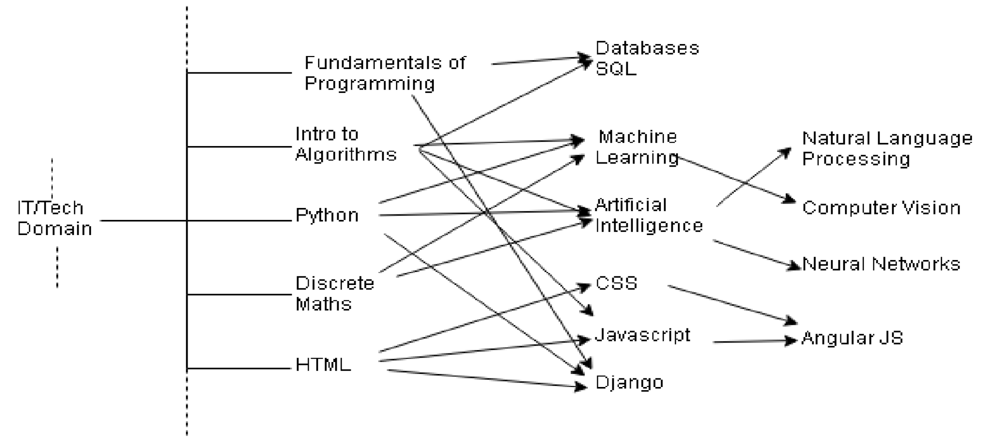


Fig. 4. A partial representation of the Skill Ontology

1) Skill Ontology: In order to extract skills from each of the segments, skill extraction module needs a skill ontology and we have built one using a wide variety of data sources available online. Figure 4 shows a partial representation of the skill ontology. It has been generated based on terms identified as skills from Onet[3] and ComputerHope[4] and the validation was done with the help of Wikipedia page and categories of the skill along with NLU services for keyword and entity extraction. We also used academic curriculum of various Universities, MIT OCW [5], details and descriptions of MOOCs, like EdX[6], to form a skill ontology inferred from the courses and course descriptions. A course may have a prerequisite for it, followed by a description of the course. We use such prerequisites to structure the curriculum in the form of dependency graphs between courses. We then process each of the course descriptions through our skill extraction module to identify the skill keywords present and then replace the course names in the graph with the list of extracted skills to obtain a skill-level graph.

Further, we generate parent-child generalization of skills leveraging multiple resource websites which were found to be rich in skill relationships. Onet is one such resource website we use along with the acyclic graph formed through category reference of Wikipedia pages of these skill terms. Information from these websites was crawled using custom web-scrapers and was then validated manually to accurately merge all the data sources and relations. This helps in better classification of data and also to find semantic similarity by judging the distance between two synsets (a group of similar data elements). For instance, if we obtain ”Java” as a skill from the candidate profile, the ontology can generalize and associate it as an ”Object Oriented Programming Language”. This ”Object Oriented Programming Language” class includes other such skills like ”Python, Ruby, C++”. Similarly, if ”HTML” is identified as a skill, then its classified under the class of ”Web Development Language”.

The skills are also categorized on a rank of 0-2, where a skill of rank-0 indicates a basic skill which has no prerequisites, while a skill of rank-1 has a prerequisite and a post-requisite skill and skill of rank-2 has only prerequisite skills and no post-requisite skills. This ranking is for an internal classification to identify the level of skill and if a higher level of skill can be acquired. This is also used for the probabilistic analysis to see if a user may have a certain indirect(implicit) skill. As an example, in Figure 4, *Discrete Math, Python, Basics to Algorithm and HTML* are classified as rank-0 skills, since they are base skills and require no prerequisite knowledge. AI, CSS and Django are rank-1 since they have both pre-skills and post-skills and similarly Neural Networks and Angular JS, Computer Vision are rank-2 since they have only prerequisites.

This ontology works on a feedback mechanism, new skills are added to the graph as new skills are learned and as we are parsing new curriculum data. Therefore the skill ontology is a directed acyclic graph(DAG) in the representation and skill is the primary entity representing the nodes.

2) Candidate profile skill extraction : Each skill in the segment is identified and classified as a direct(explicit) and an indirect(implicit) skill. A skill is categorized as an implicit skill if it is not specifically mentioned in the candidate profile but is identified based on a probabilistic approach using the skill ontology.

If for instance the user/candidate has a skill as Neural Network, and no mention of any other skill, we traverse the skill ontology, as shown in Figure 4, to identify all possible paths that could have lead to that particular skill. If there is more than 1 path that can lead to it, we use a probabilistic graphical model to assign which path is more likely to have been traversed to obtain Neural Network as a skill. This is made relevant by adding weights to path which have other skills identified in the candidate profile. For instance ”Python”

[3]https://www.onetonline.org/
[4]https://www.computerhope.com/
[5]https://ocw.mit.edu/index.htm
[6]https://www.edx.org

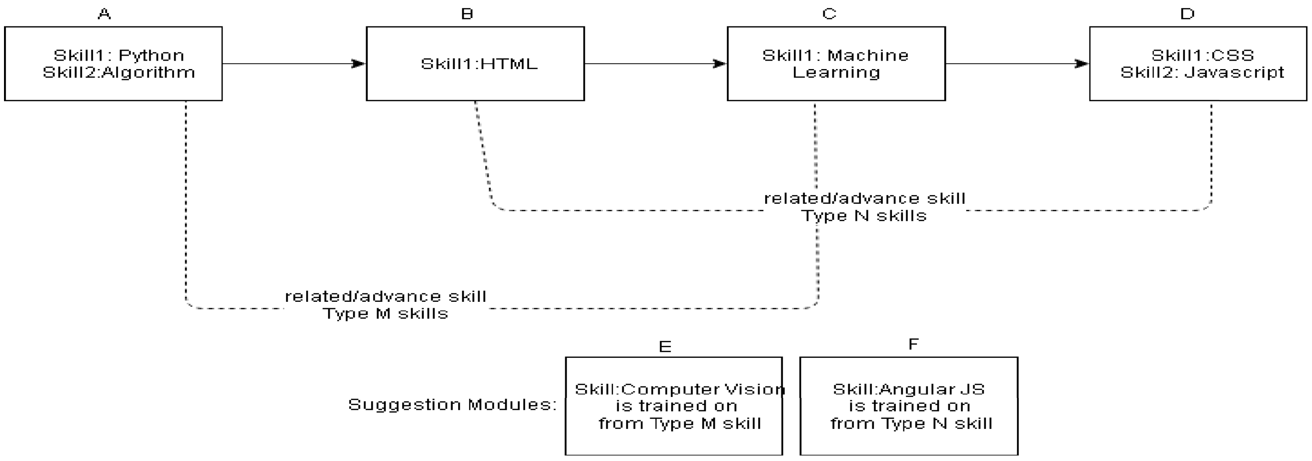


Fig. 5. Skill Graph

Score	Frequency
0	181
1	62
2	229
3	284
4	274
5	221
Total	1251
Missed	158

TABLE I
SKILL EXTRACTION STATISTICS

System	Precision	Recall	Accuracy	f1-Score
Full System	0.805	0.864	0.887	0.833

TABLE II
SKILL EXTRACTION PERFORMANCE

was also extracted as a skill and if there were 3 paths leading to "Neural Nets", the path with Python is assigned a higher probability.

In order to evaluate the performance of our *Skill Extraction module*, we extracted skills from 100 documents. These documents become our test dataset as these were selected from a corpus which was not used for training the skill extraction system. In addition these were also unseen by the developers who had developed the skill extraction model. The documents and the extracted skills were then given to four selected annotators to score the skills in terms of relevance or quality as a skill on a scale of 0-5. A score of 0-1 was given for a incorrect term that has been extracted as a skill (false positive) while 2-5 score was given based on how good the extracted skill was(true positive). We also asked the annotators to analyze and annotate skills which were missed or not extracted but should have been extracted by the system.

Table I shows the results for the analysis The Skill Extractor modules had identified 1251 total skills. Based on the annotator data, there were an additional 158 skills which the module did not identify. The respective scoring and frequency are shown in Table I and the performance is shown in Table II. The f1-score is a measure of a test's accuracy, it is the harmonic average of the precision and recall, where an f1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. We observed that the skill extraction system has a precision of 0.805 and a recall of 0.864, giving a f1-score of 0.833, which may be better performance than a related system [6]. However, it has to be noted that the lack of their evaluation dataset makes the direct comparison difficult.

3) Candidate Skills Enrichment : To build a unified skill graph that is capable of providing accurate recommendations on the next set of skills for career advancement, extracted skills are enriched with similarity, parent-child relationship and expertise dimensions. Each of these is described below:

- Similarity: Since resumes or profiles do not provide an expansive list of the skills, it is necessary to enrich skills on similarity relation. In order to find similar skills to the candidate skills, we have built a Word2Vec model [4] which is trained on Wikipedia pages of the skills in the dictionary, job descriptions from multiple job postings and course curriculum of universities.
- Parent-Child Relation : Since profiles do not mention the hierarchical relationships, it is important to enrich the skills on this dimension. For example: if Java is extracted as skill in the candidate profile without associating any hierarchical relationship with Object Oriented Programming languages, recommendation system built on such a model would miss out on the suitability of job roles for this candidate that just require basic Object Oriented Programming experience. We enrich the candidate skills with such hierarchical relationships using the skill ontology that we have developed and presented in section 3.B.1.
- Expertise: Candidate profiles also do not quantitatively express the candidate's ability or knowledge-level of skills which is needed for accurate job role recommendations. Hence, we enrich the extracted skills on expertise dimension as well.

The skill proficiency/expertise is calculated based on:

- The number of roles and positions the skill is used
- The amount of time spent using the skill (calculated as the duration of the role)
- How recently it is applied. Premise is that expertise in a skill which was applied years back would not be the same as that of a skill which is applied recently.

Using the number of roles/positions the skill is associated with, and the duration available for each of those roles, we arrive at the overall skill hours for the skill. We then apply higher weights to the recent skill hours and arrive at a normalized value for the skill proficiency. For example assume that the user had two career positions in his profile, one most recently with a duration of 1 year and the previous one with a duration of 3 years. If skill X is found in the recent position and skill Y is found in the previous position, we will associate one year skill hours for the most recent position and half year skill hours for the previous position. Thus, skill proficiency for X would be 1 year skill hours and for Y would be 1.5 year skill hours.

C. Skill Graph Generator Module

Using the extracted and enriched skill data as described in the previous sections, we generate the candidate skill graph. The output of the system is a sequence graph of the candidate based on skills. This is also a temporal sequence as each node corresponds to a set of skills extracted in a particular position or experience. The skill graph lets the user analyze different skill-paths that he could pursue. As shown in Figure 5, the candidate is identified to have 2 sets of skills classified as - Type M and Type N. The graph relays a time-variant view i.e. in the chronological order the skills were acquired in segmented data. In the 4 segments from his profile, his skills are in nodes A (Python and Algorithm), B (HTML), C (Machine Learning), D (CSS and Javascript). The graph represents that the skills of node A and C are related while that of B and D are related which are of type M and N respectively.

Using such a repository of skill graphs, a recommendation system can mine for the most relevant skill transitions for the user's current state. As shown in Figure 5, such a recommendation system could identify that with user's Type M skills, most popular next skill transition is "Computer Vision" which is associated with Job role E. Similarly, with Type N Skills, most popular next skill transition is "Angular JS" which is associated with job role F. Note that E and F job roles could be of any specific names depending on the organization. For example: E could be a *Machine Learning Engineer* or *Data Scientist* or even a *Senior Software Engineer* depending on the organization. Still the job role would be shown as one of the next positions in career path because of the conversion of career position data to skill graphs.

IV. Conclusions & Future Work

Personalized recommendations for careers, jobs and skill trainings based on candidate profile is a challenging research problem. This is specifically relevant for experience and skill intensive careers. In this paper, we have proposed a novel framework that employs text extraction techniques for generating personalized skill graph representations of candidate profile. Leveraging the graph-centric information theoretic approaches, our future work in this space will involve generating ranked recommendations on different career path options that optimally utilize the accumulated skill and experience of the candidate. We also intend to use skill graph for inferring professional growth of a user. This could help a professional in understanding where he stands when compared to his peers. Application of skill graph for professional growth inference could also help in comparing two organizations in terms of professional growth of their employees.

References

[1] S. T. Al-Otaibi and M. Ykhlef. A survey of job recommender systems. *International Journal of Physical Sciences*, 7(29):5127–5142, 2012.

[2] M. Brandmeier, C. Neubert, M. Brossog, and J. Franke. Development of an ontology-based competence management system. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 601–608, July 2017.

[3] F. Draganidis, P. Chamopoulou, and G. Mentzas. An ontology based tool for competency management and learning paths. In *6th International Conference on Knowledge Management (I-KNOW 06)*, pages 1–10, 2006.

[4] Y. Goldberg and O. Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[5] W. Hong, S. Zheng, H. Wang, and J. Shi. A job recommender system based on user clustering. *JCP*, 8(8):1960–1967, 2013.

[6] F. Javed, P. Hoang, T. Mahoney, and M. McNair. Large-scale occupational skills normalization for online recruitment. In *AAAI*, pages 4627–4634, 2017.

[7] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, H. Bersini, and M. Saerens. A graph-based approach to skill extraction from text. *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pages 79–87, 2013.

[8] M. F. Koh and Y. C. Chew. Intelligent job matching with self-learning recommendation engine. *Procedia Manufacturing*, 3:1959–1965, 2015.

[9] T. Lau and Y. Sure. Introducing ontology-based skills management at a large insurance company. In *In Proceedings of the Modellierung 2002*, pages 123–134, 2002.

[10] E. Malherbe, M. Diaby, M. Cataldi, E. Viennet, and M.-A. Aufaure. Field selection for job categorization and recommendation to social network users. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 588–595. IEEE, 2014.

[11] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[12] B. Patel, V. Kakuste, and M. Eirinaki. Capar: A career path recommendation framework. In *Big Data Computing Service and Applications (BigDataService), 2017 IEEE Third International Conference on*, pages 23–30. IEEE, 2017.

[13] T. R. Razak, M. A. Hashim, N. M. Noor, I. H. A. Halim, and N. F. F. Shamsul. Career path recommendation system for uitm perlis students using fuzzy logic. In *Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on*, pages 1–5. IEEE, 2014.

[14] H. Verma, B. Goel, and A. V. Suvarnkar. System and method for recommending personalized career paths, Apr. 1 2010. US Patent App. 12/241,497.

[15] Z. Wang, S. Li, H. Shi, and G. Zhou. Skill inference with personal and skill connections. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 520–529, 2014.