Skills Explorer: A Human+AI Approach to Data Capture for Skills

Knowledge Graph Embeddings and UI Design to Help Users Share Structured Information on Skills

Diarmuid Cahalane
Accenture Labs
Dublin, Ireland
diarmuid.cahalane@accenture.com

Justin Walker
FJORD, Accenture – The Dock
Dublin, Ireland
justin.walker@fjordnet.com

Aisling Nugent
Accenture – The Dock
Dublin, Ireland
a.nugent@accenture.com

Paul Sweeney
Accenture – The Dock
Dublin, Ireland
paul.p.sweeney@ accenture.com

ABSTRACT

We designed, prototyped and user-tested a novel approach to help users select relevant items from a large library, leveraging learned representations of the library items. We applied this to help managers select the skills they require of a worker in a new role, choosing skills from amongst those on a large reference list.

People do not naturally describe skills exclusively, exactly and verbosely in the terms available on skills reference lists. This "translation problem" was our focus as we aimed to make it seamless for people to find and list skills.

A "knowledge graph embedding" approach was used to learn contextual representations for skills from enterprise data including the lists of skills known by each employee at the client firm. This removed the need to organize skills in hierarchies or ontologies. Analytics running interactively then sift the representations to help the manager explore and discover related skills via the custom-designed user interface (UI).

In this paper we describe the system, the design of the UI, and the results of testing the system and UI with Talent Supply Chain professionals at a client organization.

KEYWORDS

Data science, knowledge extraction and discovery, human-in-the-loop, human machine interaction, human-centric design.

ACM Reference format:

Diarmuid Cahalane, Justin Walker, Aisling Nugent and Paul Sweeney. 2021. Skills Explorer: A Human+AI Approach to Data Capture for Skills. *DaSH@KDD'21, August 15-16, 2021, Virtual Conference. ACM, New York, NY, USA, 7 pages.*

1 Introduction

Shifts in skills supply and demand sometimes happen in a gradual manner, driven by forces such as emerging technologies and the increasing levels of automation in some roles and industries. The World Economic Forum estimates, for example, that by 2025 machines will do an amount of work equal to what humans will do (as measured in time spent on tasks currently) [1]. Shifts can also happen more suddenly, driven by forces such as the COVID-19 pandemic: according to the International Labour Organization, 2020 saw a decline of 8.3% in hours worked globally compared to the prior year and massive job losses in some sectors while some high-skilled sectors saw jobs growth [2]. Organizations and societies that can match people to work and training in an agile, optimized, data-driven manner stand to be more resilient. They will be able to respond to trends faster and recover from shocks more quickly. This also means minimizing economic disruption to the individuals who make up those organizations and communities, and helping them to fulfil their potential and ambitions.

The facility to articulate and reason about information on skills in terms of a shared set of skills is central to optimizing recruiting, staffing and training. Capturing data about what skills people already hold is the first step to knowing what they could do or learn next. It can also be the first part of a virtuous cycle that builds up data showing how skills relate to each other in the context of a given organization or industry. Recording the skills profiles of people, and in what roles they use their skills, generates data to better inform how skills are relevant to roles and how they are related each other. That, in turn, can inform data- and AI-driven systems that help people to articulate and reason about skills. For example, a measure of how closely skills relate to each other, sometimes called "skill proximity", can be learned from data and then used to make inferences about the extent to which holding a given skill predisposes an employee to learning proximal skills or to taking up roles wherein they can build upon that skill.

We created an AI-system and user interface (UI) to assist users with the task of selecting relevant skills from a large library of available skills. In the first instance, we designed for a user such as a project manager whose task is to select skills from the library that accurately describe those required to carry out a role. This is a "demand-side" use case (capturing skill needs). However, we envision that our system and UI can be easily adapted to supply-side use cases such as helping a user document the skills they hold or aspire to learn.

For example, and to foreshadow Section 3 below, our system can assist a user who wants to create a list of skills they think of informally as being "finance" skills. It's trivial to look up skill names in the library which contain the word finance; our skills library had entries like "Corporate Finance" and "Consumer Finance". More challenging is the task of recommending library skills that are related to "finance" but not explicitly by their names. For example, "Wholesale Lending" and "Shareholder Value Analysis". The system used learned representations of the skills to select and present additional skills which might be meaningfully related to the user's input search term.

Beyond the present application to a skills search, our system illustrates more generally how one might automatically learn representations for library items from existing data and help a user browse and discover items based on the shared contexts of those items.

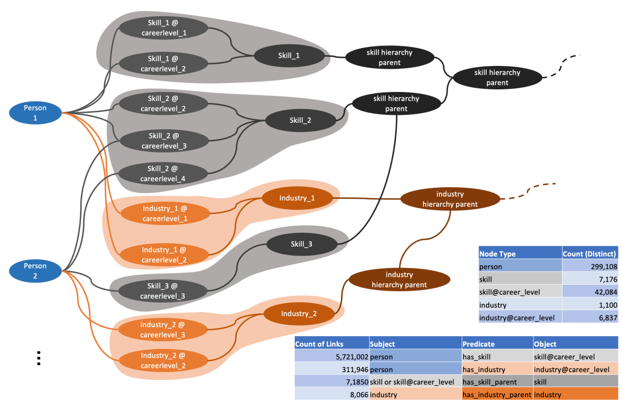
In the sections that follow we outline the solution and how we tested it. Section 2 describes the construction of a "knowledge graph" to contain input data about skills, the learning of "embeddings" (vector representations) for the skills concepts by consuming that knowledge graph, and our findings that the representations had captured meaningful contextual information about the skills concepts. Section 3 considers the design of the user interface and experience, and the associated analytics needed to make the skills embeddings useful and usable for a manager carrying out the skills capture task. Section 4 reports how we tested the usability and usefulness of the Skills Explorer with users, and the results of that testing.

2 Learning Concept Representations from Data

Enterprise databases and other repositories often contain relational data which describe how entities relate to each other, along with attributes associated to these entities. These data are verbose and do not present a ready summary or overview of a given item's context nor a view of which entities might belong to similar contexts. Knowledge graph embeddings help with both of those requirements [3].

In our case the relevant data set relates employees to the skills that they hold and to the industries in which they have experience. Summarizing which skills share similar contexts and presenting this information in a usable form could help the user discover related skills, we believed. To achieve that summarization of skills' shared contexts, we took a two-step approach. We first constructed a knowledge graph (KG) to contain the full set of entities and their relationships. The second step used neural relational learning to create a set of embeddings (vectors) representing the entities in a continuous space (200 dimensional in our case). The information which had been contained in the relationships was thereby encoded and summarized by the locations of the spatial embeddings. The vector representations are readily consumed by algorithms (e.g. nearest neighbors etc) to select and rank entities in response to user input. For any given skill, its context is thus summarized by its location in the embedding space, and it is easy to find which other skills are nearby in this space and therefore have similar contexts.

The following sections described both steps in more details and how we evaluated the output of the representation learning.



Node Type	Count (Distinct)
person	299,108
skill	7,176
skill@career_level	42,084
industry	1,100
industry@career_level	6,837

Count of Links	Subject	Predicate	Object
5,721,002	person	has_skill	skill@career_level
311,946	person	has_industry	industry@career_level
7,1850	skill or skill@career_level	has_skill_parent	skill
8,066	industry	has_industry_parent	industry

Figure 1. Schematic representation of a Knowledge Graph (KG) describing people's skills and industry experience. The counts of the distinct node and link types in the KG are shown.

2.1 Construction of a Knowledge Graph (KG)

Our KG was represented as a set of relationships in the form of "triples" which relate a subject via a predicate to an object, hence called "SPO triples" and written <*subject*, *predicate*, *object*>. There are five types of concepts which can serve as subjects and/or objects: *person*, *skill*, *industry*, *skill@career_level* and *industry@career_level*. There are four predicates: *has_skill* and *has_industry* relate people to the skills and industries (respectively) in which they have experience; *has_skill_parent* and *has_industry_parent* are for hierarchical relationships amongst skills and industries. The counts of distinct instances of each concept type are given in Figure 1, numbering over 350,000 in total. Also in the figure are the counts of the SPO triples which relate those instances to one another according to the data in the skills enterprise database. The figure provides a schematic illustration of how the SPO triples convey information including, for example, which people held which skills at which stages ("levels") in their careers: <*person, has_skill, skill@career_level*>. We also included triples to describe how the skills and industries were related by an existing hierarchy, e.g. <*industry@career_level, has_industry_parent, industy*>, <*industry1, has_industry_parent, industy2*>. It should be emphasized that the hierarchy data is not a prerequisite, but where such data are available they may help provide additional context for the subsequent machine learning steps. In total, the KG has over 6 million triples.

2.2 Concept Representation Learning

AmpliGraph is an open-source Python library comprising a suite of neural machine learning models and is used to predict links between concepts in a knowledge graph [4]. In service of such predictions, AmpliGraph learns embedding vectors for each concept in the graph, influenced by all the links between all the concepts. The vectors are optimized so that in the task of predicting links (unseen SPO triples) those which are true receive high scores whereas purposefully "corrupted" links receive low scores. Corruptions are made by swapping the subject and/or object of a true triple so that the resulting triple is much less likely to be true, e.g. <industry123, has_skill, personABC>.

For our purposes, we are not directly concerned with the output of the link prediction task, but more so with the embeddings which AmpliGraph learned to represent the concepts (in service of that task) (see Figure 2). The learned vector embeddings for similar concepts tend to be nearby in the embedding space. For us, this is useful in suggesting implicit relationships between skills and which skills may belong in similar contexts. Figure 3 shows, for example, a subset of the skills whose learned embedding vectors were nearby the embedding for the skill "Artificial Intelligence". We used AmpliGraph's ComplEx model with a dimensionality of k=100, resulting in each embedding vector comprising 100 complex numbers [5].

2.3 Evaluating the Learned Representations

How could we know whether the learned embeddings conveyed accurate and useful information about the skills? We used both quantitative and qualitative methods to address this question.

AmpliGraph has several built-in functions that evaluate model predictive performance versus ground truth data (see the AmpliGraph Docs [6] for a more detailed description than the following summary). The mean reciprocal rank (MRR) is a measure of on average how highly true triples are ranked relative to corruptions when ordered by the prediction scores for each triple. "Hits@N" gives the frequency of a true triple being in the top N when ranked by prediction score along with corruptions. What constitutes good performance relative to these metrics is subjective (according to different applications). We interpreted the values we observed for our model as indicative that the learned embeddings had successfully captured signals from the data that could contextualize skills, but that its performance on some prediction tasks for individual skills would likely be modest (MRR=0.15, Hits@1=0.12, Hits@3=0.16, Hits@10=0.22).

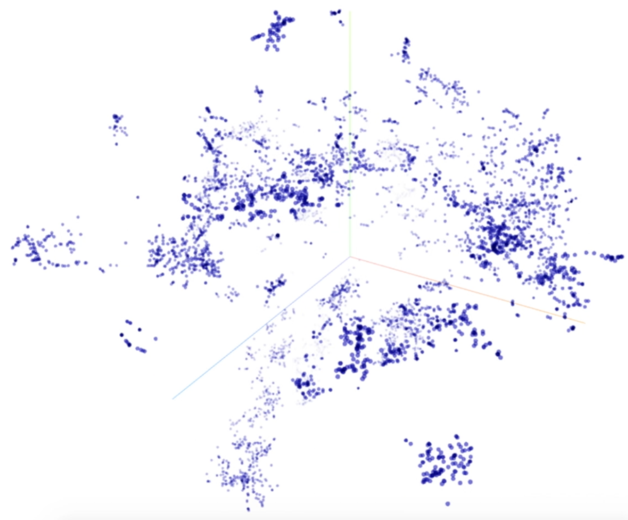


Figure 2. A three-dimensional projection (using t-SNE) of c. 6,000 skills embeddings vectors. Embeddings in a 100 dimensional space of complex numbers were learned to represent the skills in our dataset. The neighborhood relationships in the embedding space (e.g. under a p-norm with *p*=0.3) can contain skills that share similar contexts in the input data. For example, distinct islands in the figure above represented variously legal skills, IT network security skills, and digital marketing skills.

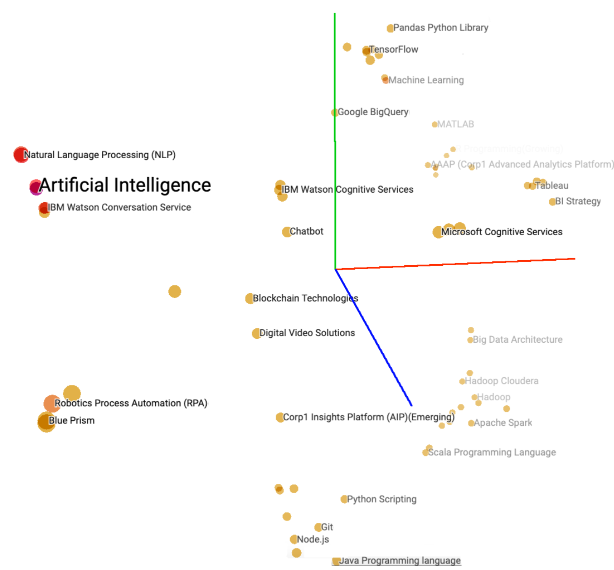


Figure 3. For the "Artificial Intelligence" skill, a three-dimensional projection (using t-SNE) of the 70 closest skill embedding vectors from among the c. 6,000 skills in the embedding space. Visualized using TensorBoard Embedding Projector [7].

Apart from the model's consistency with input data, its value derives in large part from the imputations it affords. To assess this value and gain a more "qualitative" evaluation, we asked skill subject matter experts (SMEs) to evaluate output from our model relative to a baseline model. This was in the context of a "skills substitution task". SMEs were asked to consider possible alternative skills to "substitute" for a given skill. For each of 10 target skills, both models offered 20 alternative skills and the SMEs rated their suitability on a four-point scale from 3 ("very good fit") to 0 ("un-related"). A total of 400 substitutions were rated by the experts. For 9 out of the 10 target skills, our model's suggestions were clearly preferred. The baseline suggestions were preferred in the case of one target skill. The baseline model, which the SMEs use "in production", was based on affinity analysis of the co-appearance of skills in people's resumés and on job descriptions.

Both the quantitative and qualitative evaluations suggested our model could enable tools and applications requiring skills intelligence.

3 UX Design and Prototyping: Features and Analytics

Discovering whether and how we could usefully employ the "skills intelligence" captured by the skills embeddings to help users required careful user research, UX design and additional algorithms to sift and process the embeddings in response to user input.

3.1 User Research and Design

Research was conducted across the areas of talent supply chain management and project management at the client organization. This identified a specific use case (task) that could benefit from user assistance in finding and listing skills.

The task is in the context of a processes where a user creates a "demand record" to describe a role on a project. The task of specifying skills as part of demand record creation ends when a list describing the role skills is complete. The user may or may not be deeply familiar with the specific domain of the skills.

The research enabled us to map from pain points and root causes to opportunities using selected design research methodologies including the following:

- Qualitative Interviews
- Process Mapping – As-is and To-be user flows
- Feature Prioritisation

The UI concept was prototyped and underwent a number of iterations based on review and feedback from users. The concept was brought forward with three key features and adjustments to enhance the user experience.

- Intelligent Search
- Smart Filters
- Skills Assistant

These features were co-created with stakeholders. Walkthrough testing and surveys were shared to capture feedback from users. Many iterations of wireframe designs were employed to co-create how the features might work from a user experience perspective to bring a better flow. Experimentation was aligned with analytics experiments to co-evolve the UX and search algorithms. The following sections describe the final version of the three features and the analytics underpinning them.

3.2 Keyword Matching and "Smart Filters"

The user invokes the skills search in the first instance with the familiar, "search engine-like" entry of a search term. Initial search results are presented based on a "fuzzy" string matching of the search term to skill names (based on Levenshtein distance, using [8]). Also offered at this point are "smart filter" buttons which are dynamically generated based on the skill names appearing in the fuzzy string matches. They allow the user to filter the results to show only those containing strings matching both the filter term and the search term.

The analytics underlying the filter buttons' selects terms for the buttons based on "term frequency-inverse document frequency" (TF-IDF) analysis. That is applied to the skill names returned by the fuzzy keyword search to find terms that are frequent in the search results (term frequency) but also particular to the sub-set of skills in those results more so than to the overall skills library (inverse document frequency).

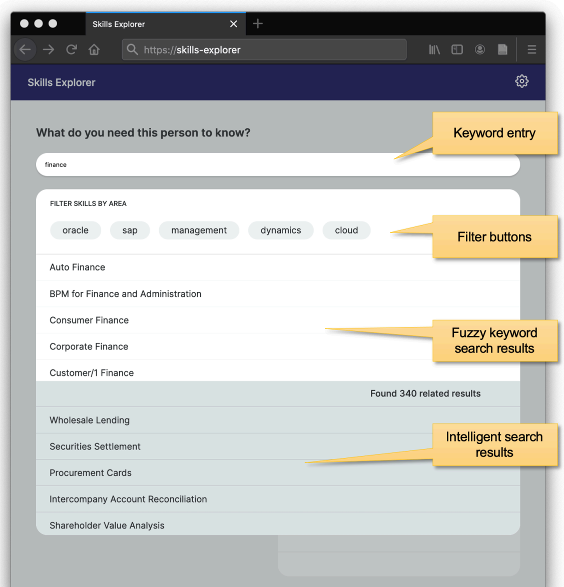


Figure 4. On entering a search term, the user is offered dynamically generated "smart filter" buttons to refine the search, results based on keyword matching and also "intelligent search" results for related skills.

3.3 Intelligent Search

In some instances, the user may find relevant skills by using the fuzzy keyword matching and filter buttons. However, those results are limited insofar as they will only contain skills whose names contain text strings matching the search term. We developed the "Intelligent Search" feature to go beyond such literal matching. By contrast, it offers the user results that are related to those skills having a keyword match but which do not have matching words or strings in their names. For example, and as seen in Figure 4, "Shareholder Value Analysis" relates to the search term "Finance" but has no matching terms in its name.

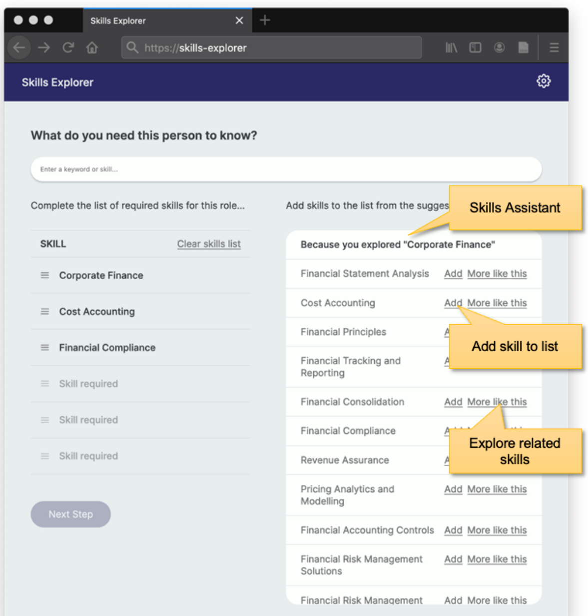


Figure 5. Having selected an initial search result, the Skills Assistant helps the user explore related skills. The user may add suggested skills to their list, explore related skills, or initiate a new search or conclude the task.

The analytics underlying this feature execute in real-time when the user opts to "Show related search results". The top matches in the fuzzy search are taken, their neighboring skills in the skills embedding space are found, ranked, and returned to the user as "related results". Skills neighbors are selected from the embedding space on the basis of applying a fractional norm to the 200 dimensional space of real numbers derived by concatenating the 100 real and 100 imaginary parts of the complex embedding vectors. We found that a p-norm having $p=0.3$ was suitable for our purpose.

When the user selects a skill from the results either the "fuzzy" or "intelligent" search, it is added to the skill list the user is populating (which can be thought of as their skills "shopping cart"). This list is seen on the left hand side in Figure 5.

3.3 Skills Assistant

Having added a first skill to their list ("shopping cart") as described above, the "Skills Assistant" is triggered (seen on the right-hand side in Figure 4). The Skills Assistant presents neighbors of the selected skill from the skills embedding space on the basis that they may also be relevant for the user to consider. The user may opt to add these related skills the list ("Add" option) or may explore related skills in the Skills Assistant panel ("More like this" option).

In due course, the user may initiate a new search by entering new search terms or may complete the task if they are satisfied with the list they have created. In practice, this data would then form part of the "demand record" for the role the user is documenting, and the user would move on to describing other dimensions of the role.

In summary, the combination of features allows a user to get started, by jumping into the space of skills via the search term to skill name string matching, and then to navigate around the space using the skills' neighbors as identified by applying analytics to the embedding vectors.

4 User Testing

Usability testing was carried out with 10 SMEs from the Talent Supply Chain team at our client organization. User feedback was positive with users in agreement that deploying the prototype would significantly improve data accuracy, user experience and efficiency. We describe the test protocol and results in more detail in the following sections.

4.1 Test Protocol

Each of ten users was introduced to the prototype by a member of the design team on a 30-minute video call with screen sharing. Each user was first given a live demonstration of the features. The user was then invited to provide search terms for skills examples they wished to examine and to direct the designer to operate the tool as they chose (choosing to explore and add skills etc). The users did not drive the interactive tool themselves simply because it would have taken a significant amount of time to provision them with access to the secure application environment.

Feature	Dimension	Survey Prompt	Scale	Average Rating
Keyword Search	Accuracy	The results of the initial keyword search were accurate.	0-Disagree to 5-Agree	4.6
Filters	Accuracy	The filters areas are relevant to my search term.	0-Disagree to 5-Agree	4.2
Filters	Usability	The filters made it easier to find a skill.	0-Disagree to 5-Agree	4.7
Intelligent Search	Accuracy	The related/intelligent search results were meaningfully related to my search term.	0-Rarely to 5-Typically	4.4
Intelligent Search	Usability	The related/intelligent search feature was challenging to use.	0-Disagree to 5-Agree	0.5
Skills Assistant	Accuracy	The related skills presented by the Skills Assistant were meaningfully related to skill I explored.	0-Rarely to 5-Typically	4.4
Skills Assistant	Usability	The Skills Assistant feature was helpful.	0-Disagree to 5-Agree	4.7
Overall	Utility	This tool helped me to create a list of relevant skills.	0-Disagree to 5-Agree	4.8
Overall	Utility	I would prefer Skills Explorer to the existing tool available to me.	0-Disagree to 5-Agree	5.0

Figure 6. Prompts and results from a survey of 10 Talent Supply Chain users at regarding their impressions of the accuracy, usability and utility of the Skills Explorer for the "demand record creation" use case. All users provided ratings for each of the prompts.

At the end of the call, users were asked to give their impressions of the prototype by responding to a survey. Prompts on the survey aimed to establish the degree to which the features were usable, provided skill suggestions that were perceived as accurate, and were useful for the completing the task. See Figure 6 for a sample of the prompts. Given that the value of the tool depends on both the quality of its suggestions (accuracy) and the quality of the design (impacting its usefulness and usability), it was important to isolate both dimensions in user testing, so as to know in which dimension any improvements might be needed. Figure 6 indicates which of the dimensions the survey prompt aimed to examine.

4.2 Results of Testing

It was apparent from the responses to the survey that the group of 10 users clearly agreed the Skills Explorer was (a) useful in the context of their task, (b) easy to use and (c) offered suggestions that were relevant and meaningfully related to their search terms. One user told us "[I] love the demo, [and I] would not make any change if it means it can be available in production tomorrow". Users rated the accuracy slightly lower than they rated the usability and utility, therefore any time invested to improve the tool might focus on that dimension in the first instance.

5 Conclusions and Future Work

We believe this prototype embodies a successful example of using a "human+AI" approach to completing a task. The human has a conceptualization of at least some of the items they mean to describe, and an understanding of the business and operational context. However, the human is limited in their ability to have awareness of vast numbers (possibly tens of thousands) of items available, to remember precisely how they are identified (by name or otherwise) in the library, and to consider the myriad possible relationships existing between the items. Those limitations are ones where they AI finds its strength. But the AI does not have the awareness of the context and understanding of requirements that the human operator brings to bear on the task. It remained, then, to create the design and data processing to make such affordances from the AI usable for the human. Our conclusion, based on development and user testing, is that our system is both useful and usable in the task of identifying and listing skills.

In the context of search, matching and discovery, we envision extending such a system to also provide "guidance" to the user on what item they might select based on considerations in addition to relatedness or proximity of items. Taking an example based on skills, it may be useful to convey signals and cues regarding factors such as demand forecasts for skills, or which skills the company management has identified as being of strategic importance [7]. In this manner, a user might be informed in a single interface about what is both proximal (as learned from by the embeddings) and desirable (based on an external signal or forecast from another system).

The addition of some form of explanation to accompany the recommendations in the UI may be useful in some use cases. However, and while it's not conclusive, in their feedback on the UI the present user group did not request explanations. Developers may be well served to validate the desirability and user value of such a feature if they are considering it to add.

Beyond the present application to skills in an enterprise context, it will be interesting to consider how such a framework might apply to other use cases where item context learned from relational data may help a user search for and discover of items.

ACKNOWLEDGMENTS

We wish to thank the HR and Talent Supply Chain teams at the client organization for their help input to creating and testing the solution. We are grateful to the anonymous reviewers for comments which helped us improve this paper.

REFERENCES

[1] V. Ratcheva, G. Hingel, S. Zahidi, S. Brown. The Future of Jobs Report 2020. World Economic Forum https://www.weforum.org/reports/the-future-of-jobs-report-2020

[2] ILO Monitor: COVID-19 and the world of work. Third edition. 29 April 2020 https://www.ilo.org/wcmsp5/groups/public/---dgreports/---dcomm/documents/briefingnote/wcms_743146.pdf

[3] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. Procs of the IEEE, 104(1):11–33, 2016.

[4] L. Costabello, S. Pai, C.L. Van, R. McGrath,N. McCarthy, P. Tabacof. AmpliGraph: a Library for Representation Learning on Knowledge Graphs (Mar 2019). https://doi.org/10.5281/zenodo.2595043

[5] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In International Conference on Machine Learning, 2071–2080. 2016.

[6] L. Costabello et al. AmpliGraph Docs. https://docs.ampligraph.org/

[7] TensorBoard Embedding Projector, https://github.com/tensorflow/tensorboard

[8] FuzzyWuzzy – fuzzy string matching in Python. https://github.com/seatgeek/fuzzywuzzy

[9] D. Cahalane, C. Guéret, C. Le Van, P. Moreau, D. Byrne: Knowledge Graphs for Workforce Reskilling Guidance. ISWC (Demos/Industry) 2020: 404-406