Estimating Fungibility Between Skills by Combining Skill Similarities Obtained from Multiple Data Sources

Shrihari Vasudevan¹ · Moninder Singh² · Joydeep Mondal³ · Michael Peran⁴ · Benjamin Zweig⁴ · Brian Johnston⁴ · Rachel Rosenfeld⁴

Received: 18 December 2017 / Revised: 21 August 2018 / Accepted: 8 September 2018 / Published online: 22 September 2018
© The Author(s) 2018

Abstract

This paper proposes an approach to estimating fungibility between skills given multiple information sources of those skills. An estimate of skill adjacency or fungibility or substitutability is critical for effective capacity planning, analytics and optimization in the face of changing skill requirements of an organization. The proposed approach is based on computing a similarity measure between skills, using each available data source, and combining these similarities into a measure of fungibility. We present both supervised and unsupervised integration methods and demonstrate that these produce improved outcomes, compared to using any single skill similarity source alone, using data from a large IT organization. The skills' fungibility matrix created using this approach has been deployed by the organization for demand forecasting across groups of skills. We discuss how the fungibility matrix is deployed to generate skill clusters and present a forecasting algorithm that additionally incorporates past/future engagements and a mechanism to quantify uncertainty in the forecast. A possible extension of this work is to use the fungibility measure to cluster skills and develop a skill-centric representation of an organization to enable strategic assessments and planning.

Keywords Skill similarity · Data integration · Fungibility · Demand forecasting · Skill-capacity analytics

1 Introduction

People-driven organizations tend to rely on an employee-centric organizational structure. The representation can enable the lookup of employee position and associations within the hierarchy. Employee skills are directly or indirectly encoded in many different information sources

ranging from their CVs to skills-sets and projects associated with them within the organization. As a result of this, a full understanding of the skills available in an organization is typically unavailable. Direct implications of this are that the resource (people) available for a particular skill is not known and that a measure of “adjacency” between people and skills does not exist.

Agile organizations need to be responsive to changing market scenarios. Demands from products and services constantly change; this results in changing skill-set requirements. Management in an organization needs to have resource information available at multiple levels of abstraction to facilitate capacity planning and decision making, both for immediate needs and for looking into the future. For instance, it is often easier to up-skill an existing employee with closely related skills than go through the process of hiring a new employee. These goals require a notion of fungibility between employees and in particular, their skill sets; fungibility in such a context may be defined as substitution with minimal up-skilling. Similarly, a measure of fungibility between skills allows organizations to improve demand forecasts for those skills considering skill usage data from prior

✉ Shrihari Vasudevan
shrivasu@in.ibm.com
Moninder Singh
moninder@us.ibm.com
Joydeep Mondal
jomondal@in.ibm.com
Michael Peran
mperan@us.ibm.com

¹ IBM Research, Bangalore, India

² IBM Research - T.J. Watson Research Center, Yorktown, NY, USA

³ IBM Research, New Delhi, India

⁴ IBM Chief Analytics Office, Armonk, NY, USA

engagements. This is due to the fact that the number of skills across an organization may be fairly large and many skills may occur relatively infrequently in historical engagement data. Aggregating skills into skill clusters using fungibility can enable more accurate demand forecasts, at the level of skill clusters, and provide a balance between forecast accuracy and usability.

This paper focuses on the problem of estimating fungibility between skills for use in capacity planning and, potentially, in the development of a skill-centric representation of an organization. In this work, fungibility is defined and estimated as a composite similarity measure between skills, considering every available data source on those skills. The paper presents a case study of a real-world business problem; it describes the context, data sources available, challenges in using the data to address the problem, design choices and methods employed, the outcome and its application to a real-world business application that has been deployed to the significant benefit of one largest IT organizations in the world. The approach presented in this paper is generic in that it is not organization or data source or industry specific.

We discuss some relevant work in Sect. 2. In Sects. 3 and 4, we first describe the two information sources used in this work—(1) skill descriptions and (2) people skill transitions, and then discuss our method for estimating skill similarity from these sources. We subsequently cast the estimation of fungibility (Sect. 5) as a problem of combining skill similarity matrices obtained using these individual data sources into a composite similarity measure. The matrix integration can be done in a supervised or unsupervised manner based on the available data. This approach to fungibility matrix generation is currently deployed by a large IT organization toward forecasting demand across groups of skills. To this end, Sect. 6 motivates and describes our demand forecasting algorithm, which incorporates notions of fungibility between skills, past/future engagements and uncertainty in forecasts. In Sect. 7, we present experimental results showing that the integrated results (fungibility) obtained from multiple sources are better than any individual source alone; we also describe its deployment in the context of demand forecasting by the large IT organization and show the improvement in demand forecasts over current practice. Finally, we discuss future work and offer concluding remarks in Sects. 8 and 9, respectively.

2 Related Work

The recent-past has experienced a spurt in skill-based analytics for human capital management. Many of these works leverage recent advances in text analytics and information retrieval to provide decision support systems based on employee skills. Richter et al. [20] describe one such

decision support tool that provides globally optimal workforce assignment recommendations for a given supply demand scenario; optimality is defined by a utility function incorporating different business metrics. Judy [9] describes a method to estimate a transferability measure for workers moving between occupations using occupational attributes based on publicly available structured databases of occupations; this requires manual selection of important dimensions/attributes involved (for each occupation as well as for combining dimensions for pairs of occupations). Bauer et al. [2] similarly address a related problem of developing skill associations for groups of employees through the use of co-clustering. Instead of a fixed skill-taxonomy representation, skills are extracted from enterprise and social data, and used to identify employees who have expertise in desired skills. Connors et al. [6] describe a system for providing substitutable employees against a demand. They use a structured database of employees and their attributes (e.g., skill or skill level), a demand (e.g., a project or a stage within it) that clearly specifies acceptable substitutions on relevant attributes (penalties for different substitutions) and develop a system to rank potential substitute employees, to give to a decision maker. No notion of fungibility between skills is developed; it is a ranking with multiple attributes problem. Finally, Ramamurthy et al. [18] present an approach for developing an adjacency model for skills as a weighted combination of other skills by using only historical employee skills data.

Other than [18], these approaches are generally concerned with identifying employees who have certain (in-demand) skills so that they can be considered for deployment to meet such demand; they do not attempt to develop a quantitative notion of adjacency or fungibility of skills. While Ramamurthy et al. [18] do address this problem more directly than others, their approach still does not yield a pairwise skill fungibility estimate. More importantly, it is unable to infer a signal for skills that are new, or have been recently introduced, and thus have few or no employees associated with that skill. These issues are addressed in this paper through the use of multiple skill data sources, including unstructured skill descriptions, skill similarity matrix generation from each data source and skill similarity matrix integration into a skill fungibility estimate.

The problem of integrating similarities between data (matrices), with a view of clustering entities, is becoming increasingly prevalent across domains in a data-driven world. Abdi et al in [1] present a principal component analysis-based approach to combining distance matrices into a single “compromise matrix” that represents a best aggregate of the original matrices. Lanckriet et al in [10] propose a kernel-based approach to data fusion of heterogeneous descriptions of the same set of entities. Similarity between entities is captured through a kernel representation. Multiple

Table 1 Expertise taxonomy

Expertise taxonomy level	Example
Primary Job Category (PJC)/Secondary Job Category (SJC)	Technical Services Other Technical Services
Job Role (JR)	System Administrator
Job Role - Skill Set (JRSS)	System Administrator - Windows
Skill	Perform system administration for Windows

kernel representations are combined using a convex optimization formulation solved using semidefinite programming techniques. In the context of recommender systems, Wang et al [23] unify user-based and item-based collaborative filtering approaches to overcome sparse ratings data being available; similarity fusion is done using a generative probabilistic framework that incorporates multiple conditional probabilities. The work [11] proposed the fusion of similarity matrices into a composite similarity matrix for clustering entities. The clustering problem is formulated as a non-negative matrix factorization problem, solved using iterative minimization process to minimize a cross-entropy measure. The fusion problem finds weights for a linear combination of the similarity matrices through a modified iterative minimization process which alternates between finding the weights and the matrix factorization given the other term. In [22], the authors use similarity network fusion to combine patient similarities obtained from multiple DNA/RNA expression types. Patient similarity matrices (on each gene expression) are represented as graphs which then inform each other through a graph message passing algorithm to converge into a fused patient similarity network. Recent papers that survey the state of the art in “multi-view” clustering (and more generally learning) include [5, 21, 24, 26]. These provide a taxonomy of recent approaches and broadly cover approaches based on co-training, multi-kernel learning, graph clustering, sub-space learning (including non-negative matrix factorization and component analysis methods) and multi-task (in a joint prediction sense) multi-view clustering. The approaches explored in this work to address the similarity matrix integration problem draw motivation from the cited works; the specific methods reported in this paper were motivated by the nature of the data and the availability of only very sparse a priori data on fungible skills.

3 Skills Data

This paper uses people skill data and skill representations of one of the largest IT organizations in the world to develop its thesis. The organization is people and skill driven. It has an employee-centric organization; however, it also has an established skills taxonomy, one particular level of which is actively used by management in decision making. This taxonomy is briefly described in Table 1 with an example.

The JRSS level of this hierarchy is particularly important to decision making—it is the functional unit of skill measurement for the organization in consideration, i.e., skill-capacity decisions are taken at this level. JRs, JRSSs and skills have descriptions that provide more elaborate details on them. Every JRSS therefore has an associated JR, JR description, JRSS description, associated skills and their skill descriptions. Whereas the existence of a skills taxonomy lends additional structure and provides information that may be useful in other contexts, it is not a requirement for the approach presented in this paper; the approach only requires the existence of a functional unit of skill measurement (hereafter, referred to simply as skill) and descriptions associated with skills; the more detailed the description, the richer the data source. As an unstructured data source (descriptions), particular challenges posed by it include sparsely populated descriptions and the existence of multiple forms of the same words (e.g., abbreviated and long forms of the same word). Skill names and descriptions (unstructured data) are one of two data sources used in developing the approach in Sects. 4, 5 and in the experiments presented in Sect. 7.

In any organization, from a skills perspective, people generally grow with time, i.e., they acquire new skills. Employees may have joined with a certain set of skills. With experience, the employee may transition to new roles or take on new projects, thereby acquiring new skills. By viewing historical data of each employee, it is possible to identify skill transitions that have occurred. Such skill transitions may represent fungibility between the skills. Potential issues with the availability of this data include (a) incomplete employee information—not all employees of all business units have up-to-date records, (b) imbalance—due to constant change in workforce in response to changing skill requirements, certain skills have only very few employees associated with them, whereas others have many and (c) skills information of employees is dispersed in many different databases. For this paper, the data are available from a single database and are assumed to be largely complete. Skill transitions captured from historical employee skill records (structured data) is the second data source used in developing the approach in Sects. 4, 5 and in the experiments presented in Sect. 7. The two data sources mentioned are specific to the organization that is the subject of this case study. In other organizations,

Fig. 1 Similarity from skill descriptions

there may be other databases or data sources available. The approaches presented can be adapted suitably to factor them.

4 Estimating Similarity Between Skills

This section discusses approaches to estimating skill similarities given skill descriptions and people skill data. The basic idea is to encode the data source (skill descriptions or people skill data) into features that enable the computation of a similarity measure between skills. Based on the structure of the data, established methods from the literature such as word2vec [16] or TF-IDF [14, 15] have been used; these choices are not requirements and alternative feature options may be used.

4.1 Skill Similarity from Skill Names and Descriptions

Every skill has associated key words and descriptions (see Sect. 3). A similarity measure between the descriptions associated with two skills could provide a valuable measure of semantic similarity between them. A recent contribution from Mikolov et al. [16] (word2vec) proposed a neural network-based approach to learn vector representations of words to capture semantic context (co-occurrence of words). The authors proposed two methods requiring opposite implementation architectures—continuous bag of words (CBOW) and skip gram (SG), to learn word vectors. The difference between the approaches is in the input–output combinations and their physical interpretations. Both approaches operate on strings of words, the input text being a collection of strings of words. The CBOW approach learns to predict a word given its context (co-occurring) words. The SG approach, on the other hand, learns to predict context words for a given word. After the learning stage, each word has a vector representation. Similarity between words can be computed by a cosine similarity between their corresponding feature vectors. The referred paper demonstrated the ability of this approach to capture syntactic and semantic

regularities in text. This paper uses the approach to compute semantic similarity between skills (names and descriptions) as shown in Fig. 1. Word-vector representations of the skills are learnt by using the word2vec skip-gram approach that learns to predict words that co-occur with the given skill word. Semantic similarity between two skills was then computed as the cosine similarity between the representations of every pair of skills; the result is a similarity matrix.

4.2 Skill Similarity from People Skill Data

Skill similarity from people skill data is computed based on the notion that if an employee has two skills, then a person may be trained from one to the other. Skill transition data are obtained from historical employee skill records as described in Sect. 3. As such, we look at each skill and compute a profile for that skill consisting of the union of all skills possessed by all employees with that skill. Then, similarity between the profiles of two skills gives us a measure of the adjacency between them.

To compute this people skill similarity, we use the term frequency–inverse document frequency (TF-IDF) [14, 15] representation to encode skills; the TF-IDF is a standard feature representation technique used in document retrieval problems. The term (TF) represents the frequency of the occurrence of a token (e.g., term or word) within a document, and is a measure of how common it is in the document. The inverse document frequency (IDF) is a measure of how often that token occurs in a corpus of documents, and is a measure of its discriminating power between documents. The product of the two terms (TF-IDF), thus, provides a measure of the importance of a token to the document, in a corpus of documents. High TF-IDF suggests that a term occurs frequently in a document but in few other documents; it could be used to distinguish a document from other documents. Every document is represented as a feature vector comprising of TF-IDF measures of tokens (e.g., words or terms) within that document. Similarity between documents is computed as the cosine similarity between their respective TF-IDF feature vector representations.

Fig. 2 Similarity from people skill data

Estimating skill similarity using people skill data is done using the TF–IDF approach as shown in Fig. 2. For every skill S , a “document” is constructed, consisting of every skill (“token”) that a person with skill S has transitioned from. Then, the term frequency (TF) for skill T in that document represents the number of people with skill S who also have skill T . Similarly, the IDF for skill T measures how common the skill T is across people with all skills. Thus, every skill S is represented by a TF–IDF feature vector representation, and skill similarity is computed as the cosine similarity between the feature vectors. This similarity encodes the historical people-based evidence in the organization. For a given set of skills, a similarity matrix is thus obtained.

5 Estimating Fungibility Between Skills

This paper defines and estimates fungibility (or substitutability) between skills¹ as the composite similarity measure obtained by combining skill similarities obtained from multiple information sources. There is a direct connection between skill fungibility and up-skill-ability or skill substitutability in practical business applications—higher fungibility increases the likelihood of a successful up-skill or substitution with minimal cost. There are two levels to this argument—(1) skills and (2) people with those skills. At the level of skills, this paper is of the view that given enough data sources, fungibility provides the most comprehensive estimate of up-skill-ability or skill substitutability. The current paper operates at this level; two informative data sources are used but the approach is not limited to these and can incorporate other data sources toward obtaining a better fungibility estimate. Most organizations tend to plan skill

requirements (vs cost) at the skill level. At the level of people (with the skills in question), skill fungibility alone does not fully determine up-skill-ability or skill substitutability as subjective or soft skill or other un-recorded factors come into play that are considered by the decision maker when taking a decision on a particular substitution or up-skilling. Our paper does not address this problem but aids the decision maker with one critical input—fungibility between skills.

Two sources of skills information are considered in this paper—skill descriptions and people skill transitions. Pairwise skill similarity matrices obtained from these sources are thus integrated into a single measure of fungibility or substitutability between skills, also a pairwise similarity matrix between skills. If the skill similarity matrix from skill descriptions is denoted by S_d and that from people skill transition data are denoted by S_p , then the fungibility matrix denoted by S is a function F of these input matrices.

$$S = F(S_p, S_d)$$

This function F may be a linear summation of the inputs or may be a more complex function. The method that determines the specific function is strongly influenced by the data that is available, in the specific context. For instance, where abundant labeled exemplars (fungible skill pairs) are available, the function could be a classifier. When data and particularly labeled exemplars are scarce, a linear summation of similarity matrices may be a very reasonable option. This work is an example of such a case, where labeled data are scarce (see Sect. 7) and hence, the function is taken to be a linear summation of the two input matrices, that maximizes an objective; where labeled exemplars are available, the objective is to produce a fungibility matrix S that maximizes the occurrence of those exemplars and where no labeled exemplars are available, the objective is to produce a fungibility matrix S that maximizes information captured from the input matrices. These scenarios and methods are discussed in the sub-sections that follow.

¹ Unlike the dictionary definition of fungibility, two skills are not freely exchangeable or there would be no distinction between them; fungibility in a skills context represents substitution with minimal time or cost or re-training.

Fig. 3 Combining skill similarity matrices into a fungibility matrix

Future extensions to this work may include the inclusion of additional skill data sources or metrics such as number of years, type of experience (on-site, off-site) to get a better estimate of fungibility between skills. Some of these may produce additional similarity matrices between skills; some may be better suited as constraints within a specific application problem. Based on deployment feedback and the availability of further fungible skill-pair exemplars, more complex functions F and methods to learn them may also be explored. The resulting fungibility matrix can in turn provide the basis for any subsequent clustering of skills to develop skill-centric representations of an organization and for applications like short-term capacity planning, strategic planning, shortlisting of candidates for re-skilling and other skill-based analyses. Combining similarity matrices can be done in a supervised or unsupervised manner depending on the data that exist. A visual depiction of this process is shown in Fig. 3.

5.1 Unsupervised Integration Using PCA

An unsupervised approach to similarity matrix integration (into a single measure of fungibility) is useful when no prior subject matter expert (SME) provided information on fungible skills or weights for the different similarity measures exist. In such cases, a reasonable objective of similarity matrix integration would be to maximize information

captured in the resulting matrix, of the input similarity matrices. Principal component analysis (PCA) provides a way to achieve this objective. Information integration in this unsupervised case is effectively treated as a dimensionality reduction problem.

PCA is a linear orthogonal projection of data; the projection produces components that greedily maximize the data variance (a measure of information in input) that they account for. Thus, the projection of the data along the first component captures most of the variance in the data; the projection along the second component is the next most informative, in this respect, and so on. Integration of similarity matrices can be done by projecting the data from the two matrices, taken together, along the first principal component that will maximally capture the variance in the input similarity matrices. The PCA approach to integration of similarity matrices is thus completely unsupervised in that it does not require SME input and relies only on the principle of information (variance) maximization. The approach is approximate in that information captured by projections along the second and subsequent components are lost and the quality of integration is dependent on the amount of variance the first component is able to capture. Steps involved are presented in the algorithm that follows. A detailed presentation of the mathematics of PCA-based similarity matrix integration is provided in [1].

Algorithm 1 (PCA similarity matrix integration)

Input: Similarity matrices S_d and S_p of dimensions $[rows, cols]$

Output: Fungibility matrix S

1. Center the similarity matrices

$$S1_p = S_p - \text{mean}(S_p)$$

$$S1_d = S_d - \text{mean}(S_d)$$

2. Reshape similarity matrix inputs and concatenate into a single matrix

$$S2_p = \text{Reshape}(S1_p, [rows * cols, 1])$$

$$S2_d = \text{Reshape}(S1_d, [rows * cols, 1])$$

$$S0 = [S2_p, S2_d]$$

3. Prepare scatter matrix

$$S1 = S0^T \cdot S0$$

4. Perform Eigen-decomposition of $S1$ and obtain Eigen values and vectors. Let the Eigen vector corresponding to the maximum Eigen value be denoted by $v0$
5. Compute the integrated similarity matrix S as the first principal component projection of the concatenated input matrix

$$S = S0 \cdot v0$$

6. Reshape S to the size of the input similarity matrices

$$S = \text{Reshape}(S, [rows, cols])$$

7. Rescale the elements of S to span the range $[0, 1]$

$$S = \frac{S - \min(S)}{\max(S) - \min(S)}$$

5.2 Unsupervised Integration Using People Data Support (PDS)

This unsupervised approach to similarity matrix integration regards people (skill transition) data as sacrosanct; skill description data are used as an alternative, when people-data information about skill pairs is unavailable or unreliable. Similarity matrix integration is performed as a linear weighted sum of the individual matrices (people and description data similarities). The weights (pwt) for the skill similarity matrix obtained from people skill data are computed as follows; the weights (dwt) for the similarity matrix obtained from description data are obtained as $dwt = [1] - pwt$. Given the skill similarity matrix from people data (S_p) and the skill similarity matrix from description data (S_d), the resulting fungibility matrix is given by

$$S = (pwt \cdot S_p) + (dwt \cdot S_d) \quad (1)$$

Fig. 4 Percentile plot (0–90 shown) of employee counts per skill. Of 1351 skills, about 265 had no employee counts, about 304 had below 20 employees each and so on. The maximum employee count for a skill ranged up to 21,132 which corresponded to the 100th percentile

People skill data, while being an invaluable resource, can be scarce for many skill pairs; not everyone diligently updates their skills information in organizations. Let n_i represent the number of people with skill i , n_j represent the number of people with skill j and n_{ij} represent the number of people who have both skills i and j . The maximum number of people that can have both skills is $\min(n_i, n_j)$. Thus, a measure of people data support for a given pair of skills is given by

$$pwt_{ij} = \frac{n_{ij}}{\min(n_i, n_j)} \quad (2)$$

To trust the people data support measure only if a reasonable support of information on individual skills is available, the denominator in the expression above can be lower bounded by some desired threshold. For this work, based on an empirical plot of the number of employees with each skill (see Fig. 4), percentiles (including median) of these numbers were chosen as the lower bound. Thus, the people data support for a pair of skills (i and j) is given by

$$pwt_{ij} = \frac{n_{ij}}{\max(\text{percentile}(n_k), \min(n_i, n_j))} \quad (3)$$

where k ranges over the set of all skills. This measure effectively penalizes people data support values pwt where the maximum possible support is below a confidence threshold.

A variation of the above penalty-based PDS computation is one of a switching model between people skill transition similarities and skill-description similarities based on whether the maximum possible support exceeds a threshold. In this case, the measure of people data support for a given pair of skills is given by Equation 2 if

$\min(n_i, n_j) > \text{percentile}(n_k)$ and zero otherwise, the latter resulting in the use of description-based similarities because $dwt = [1] - pwt$. The threshold percentile value is empirically selected as percentiles of numbers of people with each skill in consideration.

5.3 Supervised Integration Using Some Fungible Skills

When SME information about fungible skills (pairs or clusters of skills) is available, a supervised approach may be used to combine similarity matrices. If sufficient input–output data exemplars (in this case, fungible skill pairs) exist, supervised classification models may provide a feasible option. As will be discussed in Sect. 7, given very few skill pairs to work with, the choice of approach had to be more conservative. The objective of similarity matrix integration in this case would be to produce a resultant matrix that maximizes the likelihood of occurrence of the exemplars (fungible skill pairs) provided by the SME.

A way of achieving this objective is to find a weighted linear combination (LINSUM) of (normalized) similarity matrices that maximizes the clustering outcome F-measure score (or similar metric), for the given set of exemplars. The search for weights can be done using an off-the-shelf optimization tool. Bounds on weights and constraints (e.g., sum of weights = 1) can significantly simplify the search process. An outline of the steps involved follows.

Algorithm 2 (LINSUM similarity matrix integration)

Input: Similarity matrices S_p and S_d of dimensions [rows, cols] and SME provided valid skill clusters C representing fungible sets of skills; the skills covered in C are denoted by α .

Output: Fungibility matrix S

1. Compute normalized similarity matrices (range = [0, 1]) for both S_p and S_d

$$S_p = \frac{S_p - \min(S_p)}{\max(S_p) - \min(S_p)} \quad \& \quad S_d = \frac{S_d - \min(S_d)}{\max(S_d) - \min(S_d)}$$

2. Extract sub-matrices of the similarity matrices containing only skills referred to in the SME exemplars. The next step operates on only the sub-matrices to estimate weights.

$$s_p = S_p[\alpha] \quad \& \quad s_d = S_d[\alpha]$$

The above operation recovers indices for skills α from the set of all skills and extracts rows and columns with those indices.

3. Let w be the set of all weights in the search space of weights. For, e.g., if the sum of weights = 1 and weights

are discretized to a single decimal place, the set of possible weights would be $\{(0.0, 1.0), (0.1, 0.9), \dots, (1.0, 0.0)\}$

4. For each (w_p, w_d) in w
 - (a) Compute a linear weighted sum of similarity sub-matrices to obtain the fungibility matrix S

$$S' = w_p \cdot s_p + w_d \cdot s_d$$
 - (b) Given S' , use an off-the-shelf clustering approach [25] to obtain skill clusters C' —e.g., partitioning around medoids (PAM) or hierarchical clustering. Number of skill clusters is set to the number of SME provided exemplar skill clusters; this could be used to pick the clustering level if hierarchical clustering is used.
 - (c) Generate skill pairs from the clusters C (SME provided) and C' (obtained from fungibility matrix). For, e.g., a cluster with (1,2) gives (1,1), (1,2), (2,1) and (2,2). Skill pairs from C may be generated once before the commencement of the current (iterative) step.
 - (d) Compute the Precision (fraction of the skill pairs from C that are also obtained from C' relative to the total number of skill pairs in C'), Recall (fraction of the skill pairs from C that are also produced from C' relative to the total number of skill pairs in C) and F-measure (harmonic mean of Precision and Recall).
 - (e) Maintain a record or table of the F-measure for the set of weights used; in an optimization tool, the cost function would be the F-measure with the objective being its maximization.

5. From the table created above, choose weights (w_p, w_d) that maximize the F-measure of the clustering outcome. Where multiple solutions are obtained, the first can be chosen; other methods can also be used, e.g., maximize sum of precision and recall and use most similar weights to balance information from different sources.

6. Compute the (full) fungibility matrix as a linear sum of the input similarity matrices with weights (w_p, w_d) ,

$$S = w_p \cdot S_p + w_d \cdot S_d$$

This work was implemented in Python. Particular libraries that have been used include NLTK [3], Gensim [19], the SciPy stack [8] and Scikit-learn [17].

6 Demand Forecasting for Groups of Skills

The notion of fungibility, as described by the fungibility matrix obtained from Sect. 5, has been successfully applied to the business application of forecasting demand

across groups of skills. This section serves to lay out the manner in which the fungibility matrix between skills is used in the context of demand forecasting and presents our forecasting algorithm. Details on the deployment of the fungibility computation and the demand-forecast application in a large IT organization are presented in Sect. 7.

Basic demand forecasting methods essentially rely on some form of averaging over historical demands of various skills. Advances in this area have focused on improving the forecasting algorithm or outcomes through the use of hybrid multiplicative/additive models [13], the use of skill-categories (keywords) for forecasting [13], the incorporation of different factors like the status of ongoing engagements, potential signings [4]. The work [6] required explicit specification of acceptable substitutes (and associated penalties) from the demand side.

While these advances are all relevant, this paper extends the state of the art through the incorporation of multiple novel features within the proposed forecasting approach. These include (1) the use of skill clusters obtained using the previously described (see Sect. 5) automated method to estimating fungibility between skills, given multiple skill information sources, (2) the use of both historical engagements and those in the pipeline and (3) the factoring of uncertainty, to produce forecasts with confidence bounds, through a simulation of successful outcomes of engagements which are in the pipeline.

Given the expertise taxonomy in Table 1 and given that JRSSs are the functional unit of skill measurement for the IT organization being studied in this paper, demand forecasts at the level of individual JRSSs are generally most useful to a business decision maker. However, the lack of historical data often results in inaccurate outcomes. Methods such as forecasting skill-categories as demonstrated in [13] or highly abstract classes of skills (e.g., at the JR level) may provide more accurate forecasts due to availability of more data but are typically not useful for a business decision maker due to the abstract nature of the forecasts; these abstractions do not capture a function-level similarity between skills that is critical to enabling meaningful substitutions within a capacity planning and optimization scenario. This work proposes a trade-off between accuracy and usefulness of the forecasts by grouping JRSSs into fungible clusters and estimating forecasts at the level of these clusters. The algorithm used to produce these forecasts is presented below; JRSSs are replaced with the word “skill” following the taxonomy-agnostic notation used throughout this paper. A more detailed description of proposed demand forecasting approach, that includes the rationale for each step and implementation details, is presented in [7].

Algorithm 3 (Demand forecasting across groups of skills)

Inputs: A list of skills and a fungibility matrix S between them, obtained as shown in Sect. 5.

Output: Demand forecast with confidence intervals for each skill cluster.

1. The fungibility matrix is used to cluster skills into skill clusters.
2. A dataset of skill-cluster shares (columns) for historical opportunities (rows) is created.
3. K-Means clustering is performed on these shares; the number of clusters is empirically chosen (e.g., 17 in our case). These clusters are called labor profiles.
4. A weighted Multinomial-Logit model is trained using historical opportunity features (e.g., revenue, duration) to predict the above labor profiles.
5. The weight is based on claim recency (a time decay function) and size (total number of claim hours); an opportunity with many claims that recently occurred is given the most weight. This is because recent and large claims are more likely to decide the labor profiles needed for future opportunities.
6. The trained model is applied on pipeline opportunities (represented by features like revenue, duration) to predict the probability of the opportunity using the above (e.g., the 17) labor profiles. Weights are not used for this step.
7. A dot product of the predicted probabilities with the weighted average shares of each labor profile provides an estimate of predicted share of skill clusters per pipeline opportunity.
8. Using a separate linear model between expected revenue and hours, total hours for each pipeline opportunity is obtained. When multiplied by the above predicted skill-cluster shares, predicted hours per skill cluster per potential opportunity is obtained.
9. A Monte Carlo simulation uses win probabilities for pipeline opportunities (outcome of a separate project) to factor uncertainty in their labor requirements. Hundred thousand simulations assign 1 or 0 to each pipeline opportunity based on its win probability. The product of these simulations with the expected hours and a summation of the hours for each skill cluster gives hundred thousand possible expected hours for each skill cluster. This distribution is used to derive a demand forecast with confidence intervals for each skill cluster.

7 Evaluation and Deployment

A set of 1351 skills of a large IT organization were considered. As mentioned in Sect. 3 and Table 1, each skill had associated meta-data, capabilities and all of their

descriptions associated with it. These were subject to preprocessing to handle word-variations (short forms), typos, singular/plural word forms, uninformative words, symbols and punctuation marks. Preprocessed words were concatenated into lists of words with the skill word as an additional identifier/tag word for each such list. These data were used as input to learn word2vec feature vectors for the skill tag words. The representation basically learns to predict co-occurring words to the skill word in question. Given feature vectors for each skill, pairwise similarities between them can be computed using the cosine similarity between the corresponding feature vectors. The result is a similarity matrix obtained from skill descriptions.

Employee records of over 400 thousand employees were mined to extract their current skills and all skills associated with them until the current time. A percentile plot of employee counts per skill is shown in Fig. 4. The figure shows 0–90th percentile to clearly show the distribution across a majority of skills. About 265 skills had no recorded employee counts and about 304, below 20 employees each. Among the 1351 skills in consideration, the maximum employees for any one skill ranged up to 21,132—corresponding to the 100th percentile.

A TF–IDF feature vector representation for each skill was learnt in terms of all skills in question; the TF and IDF values for each token were decided by the employee

skill records. As before, given feature vectors for each skill, pairwise similarities between them were computed using the cosine similarity between corresponding feature vectors. The result is a similarity matrix obtained from employee skill records.

Similarity matrix integration was then performed using the PCA and LINSUM approaches. The LINSUM integration was initially carried out using a preliminary SME feedback of 8 groups of skills obtained through earlier discussions; the 8 clusters covered about 30 skills of the total of 1351. A sample of these clusters is shown in Fig. 5. A single skill cluster (obtained separately from each similarity matrix) depicted in Table 2 showed a similarity between skill clusters obtained from each information source used. Note the similarity in the clusters generated in Table 2 and the first cluster provided by the SME in Fig. 5.

The resultant fungibility matrix was then used to cluster the skills into about 462 groups of fungible skills (the number was chosen empirically based on the groups produced) which were presented to the business; these were very well received. At this stage, the quality of the skill clusters generated using both approaches was similar; thus, the decision was made to deploy the PCA-based clusters as the business felt using the unsupervised approach made more sense, given the limited amount of feedback available initially.

Fig. 5 Sample SME provided skill clusters. The figure shows 6 clusters from the first batch of 8 skill clusters obtained through discussions with SMEs; 8 such clusters covered about 30 skills of the total of 1351; very few fungible skill pairs were available to us. The # represents one or more proprietary products of the organization being studied in this paper

Table 2 A single skill cluster showing similarity between outcomes from skill similarity obtained from people skill transitions and from skill descriptions

People skill transitions	Skill descriptions/semantic similarity
Systems Administrator: VMWare	Systems Administrator: VMWare
Systems Administrator: Windows	Systems Administrator: Windows
Systems Administrator: Citrix	Systems Administrator: Citrix
	Systems Administrator: # Linux
	Systems Administrator: Unix Operating Systems AIX
	Systems Administrator: Unix Operating Systems HP
	Systems Administrator: Unix Operating Systems Linux
	Systems Administrator: Solaris

The # represents a proprietary product of the organization being studied in this paper

Upon presentation of these groups of skills to the SMEs, additional feedback collection was also started. The feedback expected was in the form of comments against each skill, listed by group. SME feedback obtained included blank lines (no comments) and explicit comments grouping specific skills together. Blank lines (i.e., no comment) may be assumed as approval. However, given that SMEs had to parse 1351 skills and group numbers, blank lines may also reflect the case of partial feedback for the most obvious of skill groupings that were apparent to the SME. For the purpose of this paper, only skills accompanied with explicit feedback on their groupings (i.e., fungible) with other skills are considered for evaluation. Their comments were manually parsed and a set of 19 groups of skills (about 71 in total) were obtained. With this additional data, we could once again evaluate LINSUM against PCA as well as the other methods. The aforementioned skill groups constituted the exemplars against which different methods were evaluated. The LINSUM deployment process is shown in Fig. 6.

Similarity matrix integration was once again performed using the different methods presented in Sect. 5. The rows and columns corresponding to the exemplar skills were extracted from the fungibility matrix. Using this fungibility

(sub-)matrix, a clustering procedure was performed and the clustering outcome was evaluated with respect to the SME input. These evaluation steps are similar to step 4 of the supervised approach to similarity matrix integration presented in Sect. 5.3. The resulting fungibility matrices obtained agreed with SME inputs to the extent of the precision, recall and F-measure scores listed in Table 3. The table also shows the metrics obtained using only people skill transition data or skill description data. It shows outcomes of the people data support approach—using both a penalty-based support measure (Equation 3) and the switching model (Equation 2 or zero subject to the threshold), for three different confidence thresholds defined in terms of the employee counts per skill.

The similarity matrices from skill descriptions and people skill transition data are, respectively, shown in Figs. 7 and 8. The three best integrated fungibility matrices obtained using the LINSUM, PDS and PCA approaches are, respectively, shown in Figs. 9, 10 and 11. The LINSUM outcome

Table 3 Performance comparison—fungibility estimation methods

Integration approach	Precision	Recall	F-measure
People data similarities	0.68	0.55	0.61
Description data similarities	0.61	0.61	0.61
PCA	0.64	0.62	0.63
LINSUM	0.84	0.65	0.73
PDS 25 percentile	0.48	0.46	0.47
PDS 50 percentile	0.52	0.47	0.49
PDS 75 percentile	0.6	0.61	0.6
PDS 25 percentile switching	0.68	0.54	0.6
PDS 50 percentile switching	0.7	0.5	0.58
PDS 75 percentile switching	0.68	0.62	0.65

Fig. 6 LINSUM deployment process

Fig. 7 Similarity matrix from skill descriptions

Fig. 8 Similarity matrix from people skill transition data

Fig. 9 Fungibility matrix integrating both people and description similarities obtained using the supervised LINSUM approach

is composed approximately of about 72% people skill transition similarity information and 28% skill description information and hence appears rather different from the other two outcomes. This particular weighting was chosen as it maximized F-measure, maximized the sum of precision and recall and balanced information used from individual matrices. The inverse combination of about 26% people skill transition-based similarity and 74% skill description similarities performed almost identically well—the outcome of this case would appear similar to the PDS and PCA approaches. Provision of further exemplars of fungible skills could further narrow down the weight combinations giving best outcomes in the supervised LINSUM case. As such, we intend

Fig. 10 Fungibility matrix integrating both people and description similarities obtained using the unsupervised people data support (PDS) approach

Fig. 11 Fungibility matrix integrating both people and description similarities obtained using unsupervised PCA approach

to replace PCA with LINSUM in subsequent deployments as more feedback becomes available. Each of the three outcomes shown produced outcomes superior to that obtained using the individual similarity matrices by themselves; this demonstrates capture of synergy between the different skill similarity information sources.

Fungibility between skills was used to cluster them into skill clusters. These skill clusters are being used by the IT organization to forecast demand, given skills usage information from prior engagements and those in the pipeline. Forecasting demand at the level of skill clusters (instead of

individual skills) achieves good trade-off between accuracy and usability. This is due to the fact that the number of skills is fairly large, but many skills are rare in historical engagement data and forecasting their demand directly is not accurate. However, aggregating them using fungibility, into skill clusters, results in higher prediction accuracy.

Figures 12 and 13 show the validation test of the demand forecast. Forward validation revealed that the approach presented was superior to the status quo for

Fig. 12 Comparison of demand quantity in the updated forecast obtained using the proposed approach relative to actual demand quantity and an existing forecasting system

Fig. 13 Actual demand quantity compared to the forecast 95% CI lower bound, plotted on a log scale to enhance visualization

the large IT organization being studied in this paper. An existing system had workforce-management provide their anticipated demand for projects in the pipeline. This system was compared to the demand forecast approach presented in this paper by examining their accuracy relative to actual demand for a six-month period—this is shown in Fig. 12. In comparison with actual demand, the demand forecast from the proposed approach, based on fungibility between skills, was closer than the existing management system for 93% of the skill clusters. Additionally, a view of forecast demand compared to actual demand at the skill cluster level is available as shown in Fig. 13. A more accurate demand forecast allowed this large IT organization to increase IT labor gross profit by 2.3%. This resulted in a multi-million dollar impact due to better planning of high-demand skills and avoiding high premiums on short-notice labor resource shortages. More details on the proposed demand forecasting approach are discussed in [7].

It should be emphasized that an objective measure of fungibility between skills does not exist in the organization whose data have been used for this study; we believe this is the norm across organizations. Given the current state of no clear baseline and sparse SME inputs to work or validate models with, the similarity matrices from individual data sources already provide a measure of substitutability between skills. This paper further defines fungibility as a composite similarity score between skills, considering all available data sources in the organization; the improved outcomes from the composite measure demonstrate the potential for synergistic use of multiple skill data sources, that an organization might have, toward estimating an objective measure of fungibility between skills.

While the paper describes and evaluates both supervised and unsupervised approaches to integrating skill similarities, the emphasis of the paper is in the definition and estimation of fungibility between skills obtained by integrating skill similarities obtained from different information sources; the integration is performed using a supervised or unsupervised approach as afforded by the application scenario. Other integration methods exist and may be used in place of those presented in this paper. Quality or uncertainty measures for individual data sources may also be incorporated if there is explicit prior knowledge or a reference to compare with and a validation mechanism. The fungibility measure may be used within skill-based recommender systems; its definition in terms of multiple information sources of skill similarity lends it a level of robustness to “cold-start” problems (ill-defined or incomplete information source) that may occur in enterprises with less rigorous skill taxonomy or recording mechanisms.

8 Future Work

Numerous avenues for extending this work exist; these may consider additional data sources or evaluate alternative representation and integration methodologies. Some specific examples include the inclusion of Wikipedia (or similar resource) as an additional source of skill description information, particularly in relation to domain-specific keywords, the use of alternative vector-space representations to encode skills (e.g., paragraph vectors in [12]) and the detailed comparison of multiple methods of integrating skill similarity matrices.

9 Conclusion

A measure of fungibility between skills was defined to enable effective capacity planning, analytics and optimization for modern organizations that have to cope with ever-changing skill requirements. Fungibility was estimated as a combination of skill similarities obtained from different information sources including (but not limited to) people skill transition data and skill descriptions. Supervised and unsupervised similarity matrix integration methods were presented and experiments (with this measure) demonstrated improved outcomes compared to using any single skill similarity source alone. Fungibility matrix estimated using the unsupervised PCA-based matrix integration approach has been deployed by a large IT organization for clustering skills for use in demand forecasting. To this end, a demand forecasting approach for fungible clusters of skills that factors historical engagements, potential engagements and uncertainty over their successful acceptance, was also presented. Demand forecasts obtained were shown to be very competitive with respect to actual demand and significantly more accurate than the current approach employed by the IT organization. Feedback on the current deployment of the fungibility estimation approach has resulted in additional improvements in the performance of the supervised similarity matrix integration (LINSUM) approach which is, thus, expected to replace the unsupervised approach in subsequent deployments.

Acknowledgements The authors thank Jonathan DeBusk, Pankaj Srivastava, Owen Cropper and Steve Loehr for their support and valuable inputs. The authors also thank the reviewers of this paper for their constructive feedback.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate

credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abdi H, O'Toole AJ, Valentin D, Edelman B (2005) Distatis: the analysis of multiple distance matrices. In: IEEE computer society conference on computer vision and pattern recognition-workshops, 2005. CVPR workshops. IEEE, pp 42–42
2. Bauer J, Fang D, Mojsilovic A, Ramamurthy K, Varshney K, Wang J (2015) Method, system and computer program product for automating expertise management using social and enterprise data. US Patent 20150317376A1
3. Bird S, Loper E, Klein E (2009) Natural language processing with python. O'Reilly Media, Inc., Newton
4. Cao H, Eaton M, Hsieh M, Hu J, Li T, Ray B (2007) Method and apparatus for workforce demand forecasting. US Patent 8015043B2
5. Chao G, Sun S, Bi J (2017) A survey on multi-view clustering. arXiv preprint [arXiv:1712.06246](https://arxiv.org/abs/1712.06246)
6. Connors D, Fasano J, Gresh D (2008) Method for substitution of employees in a service engagement. US Patent 20080294486A1
7. Johnston B, Zweig B, Peran M, Wang CJ, Rosenfeld R (2017) Estimating skill fungibility and forecasting services labor demand. In: Proceedings of the international workshop on enterprise big data semantic and analytics modeling, colocated with IEEE Big Data 2017
8. Jones E, Oliphant T, Peterson P et al (2001) SciPy: open source scientific tools for Python. <http://www.scipy.org/>. Accessed 1 Apr 2016
9. Judy R (2011) Method for statistical comparison of occupations by skill sets and other relevant attributes. US Patent 8082168B1
10. Lanckriet G, Deng M, Christianini N, Jordan M, Noble W (2004) Kernel-based data fusion and its application to protein function prediction in yeast. In: Proceedings of the Pacific symposium on biocomputing
11. Lange T, Buhmann JM (2006) Fusion of similarity data in clustering. In: Advances in neural information processing systems, pp 723–730
12. Le Q, Mikolov T (2014) Distributed Representations of Sentences and Documents. In: Proceedings of the international conference on machine learning (ICML), pp 1188–1196
13. Lee Y, Connors D (2008) Method and system for forecasting workforce demand using advance request and lead time. US Patent 20080249839A1
14. Leskovec J, Rajaraman A, Ullman JD (2014) Mining of massive datasets. Cambridge university press, New York
15. Manning C, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
16. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of the international conference on learning representations
17. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
18. Ramamurthy KN, Singh M, Davis M, Kevern JA, Klein U, Peran M (2015) Identifying employees for re-skilling using an analytics-based approach. In: IEEE ICDM workshop on data mining for service, pp 345–354

19. Rehurek R, Sojka P (2010) Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks
20. Richter Y, Naveh Y, Gresh D, Connors D (2007) Applying constraint programming to workforce management of highly-skilled employees. In: Proceedings of IEEE international conference on service operations and logistics, and informatics (SOLI)
21. Sun S (2013) A survey of multi-view machine learning. *Neural Comput Appl* 23(7–8):2031–2038
22. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, Haibe-Kains B, Goldenberg A (2014) Similarity network fusion for aggregating data types on a genomic scale. *Nat Methods* 11(3):333
23. Wang J, De Vries AP, Reinders MJ (2006) Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 501–508
24. Xu C, Tao D, Xu C (2013) A survey on multi-view learning. arXiv preprint [arXiv:1304.5634](https://arxiv.org/abs/1304.5634)
25. Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. *Ann Data Sci* 2(2):165–193
26. Yang Y, Wang H (2018) Multi-view clustering: a survey. *Big Data Min Anal* 1(2):83–107