

METHOD FOR MATCHING POTENTIAL JOB APPLICANTS TO A JOB OPENING

The invention relates to talent management and resource allocation. In particular, it relates to computer-based techniques for calculating scores for fresh graduate to determine their suitability for a full time or internship position, and a platform to support such techniques.

In today's globalised economy with large number of projects and dwindling budgets, project managers are tasked with allocating the right resources to a project that will ensure maximum productivity for employees which would ultimately lead to improved bottom-line for the companies.

Resource allocation is a key challenge in any organisation or company. Currently, a lot of resource allocation of employees/interns to projects is performed manually based on human judgement. Not only is it time consuming but it also introduces the risk of unconscious bias when allocating resources. When it comes to resource allocation, employees are usually in one of the two states: under-utilized or overworked, neither of which is ideal for a company. Resource allocation makes sure that such kind of productivity imbalances can be eliminated and scarce human resources are utilized effectively.

The platform manages an inventory of job openings and candidate applications for each job. The platform is highly customizable and factors like number of jobs seen by candidates or the maximum number of candidate applications received by a company can be tweaked as per requirement. Within these constraints, the platform recommends appropriate matches to improve job opening welfare or candidate welfare or maximum matchings depending on the end goal.

With 10 candidates and 10 jobs, the first (or top) candidate would match with 10 jobs, while the second candidate matches with 9, and so forth. This results in 3.6 million possible permutations which is very difficult to find an optimal solution. With 50 candidates and 60 jobs, this equates to 2.3×10^{75} permutations which would be near impossible to find an optimal solution by a human. Hence, there is a need to provide a method and system that

can provide an optimal solution of the matching of the candidates and jobs in a short time, and not a random or a “first come first serve” solution.

5 The listing or discussion of an apparently prior-published document in this specification should not necessarily be taken as an acknowledgement that the document is part of the state of the art or is common general knowledge.

Any document referred to herein is hereby incorporated by reference in its entirety.

10 In a first aspect of the invention, there is provided a computer-implemented method for matching a potential candidate desirous in seeking employment to a job position, the method comprising:

- (a) providing a database comprising job requirements for at least one job position and at least one candidate profile data set;
- 15 (b) computing a job score for a candidate for each job position by comparing the candidate profile data set with said job requirements; and
- (c) generating a set of job recommendations for each candidate based on the job score by a minimum cost flow method.

In an example, the set of job recommendations may contain from one to ten job
20 recommendations.

Preferably, the database comprises at least one job position with a job description, and the method comprises analysing said job description to identify the job requirements.

25 Preferably, the job description is analysed by parsing the job description and matching the parsed data to a set of abilities data to identify job requirements associated with the job position.

Preferably, the database comprises at least one candidate and the method comprises
30 creating the candidate profile data set associated with said candidate.

Preferably, the candidate profile data set comprises at least one selected from the group comprising of skills competency, skills to learn, courses taken, academic results, leadership qualities, involvement in volunteer work, the candidate's preferred industry, the candidate's preferred job function, the candidate's location, the candidate's expected salary, the candidate's company type preference, and psychometric data of the candidate, and computing a skills score, a skills to learn score, a course score, an academic results score, a leadership score, a volunteer score, an industry score, a job function score, a location score, a salary score, a company type score, and a psychometric score respectively based on the available data in the candidate profile data set and an individual score based on the candidate's preferred job function. The candidate profile data set may be created by the candidate inputting the data into a platform or form with the relevant fields. Alternatively, a physical copy of the candidate's curriculum vitae may be parsed and analysed to extract out the relevant fields to provide the candidate profile data set.

In an example, the job score is based on an average of a candidate capability score and a candidate compatibility score. In another example, each of the candidate capability score and the candidate compatibility score is individually used as the job score to generate the set of job recommendations.

In an example, the candidate capability score is computed based on the skills score, the course score, and the leadership score. Preferably, the candidate capability score is further computed based on at least one of the academic results score and the volunteer score.

Preferably the candidate capability score is computed based on weightages in the job requirements. This allows the hiring company to choose which components are more important based on the hiring company own organisation metrics, and allows for customisation of the job requirements for each company.

In an example, the skills score is computed by iterating each skill in the job requirements against the skills competency in the candidate profile data set; recording a highest score generated for each skill; summing the highest scores for each skill in the job requirements to generate the skills score. Preferably, the skill score is normalised based on a ranking or

rating of the candidate's skills in the skills competency data. As an example, the skill score is normalised by combining the ranking or rating of each skill of all the candidates to generate a distribution curve; selecting multiple percentile points to map a function; and generating a normalised requirement score for each skill and used in summing the highest scores to generate the skills score.

In an example, at least one of the following conditions is fulfilled:

- (a) the course score is computed by determining if there is a first match between a course taken by the candidate and a course requirement in the job requirements; and computing the course score based on whether the first match is present and an existing employment data set;
- (b) the academic results score is computed by determining if the academic results in the candidate profile data set meet a minimum threshold in the job requirements;
- (c) the leadership score is computed by classifying the leadership qualities in the candidate profile data set into one of a plurality of groups, each group having a specific value and classified based on an existing employment data set; and
- (d) the volunteer score is determined by determining if the involvement in volunteer work in the candidate profile data set meet a minimum threshold in the job requirements.

Preferably, two or three or four of the conditions are fulfilled.

Preferably, the compatibility score is computed based on at least one of the following components: the skills to learn score, industry score, job function score, location score, salary score, individual score, and psychometric score.

Preferably, the components of the compatibility score may be weighted according to the job requirements.

In an example, at least one of the following conditions is fulfilled:

- (a) the skills to learn score is computed by iterating each skill in the job requirements against the skills competency in the candidate profile data set; recording a highest score generated for each skill; summing the highest scores for each skill in the job requirements to generate the skills score;

(b) the industry score is computed based on a similarity between the candidate's preferred industry and an industry of the job position, and a rank in the candidate's preferred industry;

(c) the job function score is computed based on a similarity between the candidate's preferred job function and the job function in the job requirements, and a rank in the candidate's preferred job function;

(d) the location score is computed by determining a commute time between the candidate's location and a location of the job position; and determining the location score based on a function derived from a plotted distribution of commute time for the job position or similar job positions;

(e) the salary score is computed based on an extent the candidate's expected salary falls within a salary range in the job requirements;

(f) the individual score is computed based on a numerical difference between the candidate's preferred job function and a categorisation of the job function in the job requirements.

Preferably, two or three or four or five or six of the conditions are fulfilled.

Preferably, a network simplex method is used in generating the set of job recommendations.

Preferably, the method further comprises generating a second set of job recommendations for each candidate based on a ranking of the candidate's preferences.

In an example, at least one of the following conditions are fulfilled:

- (a) there are at least 50 candidates in the database;
- (b) there are at least 50 job positions in the database;
- (c) there are more job positions than candidates.

Preferably, two or three of the conditions are fulfilled. Advantageously, the large number of candidates and job positions mean that the number of possible permutations is too large to be computed efficiently by a human. Further, the methods described herein avoids human bias in the allocation or matching processs, whether intentional or unintentional.

The method may further comprise receiving the at least one job position from an organisation (or company). This may be via an online portal to receive the job position and job requirements, or by receiving a hard copy and converting the hard copy into a digital version by optical character recognition and natural language processing methods. Similarly, the method may further comprise receiving at least one candidate curriculum vitae either through an online portal or via a hard copy, and creating the candidate profile data set in a similar manner as the job requirements.

10 In a second aspect, there is provided a non-transitory medium comprising instructions which when executed by a processor causes the processor to perform the method according to the first aspect.

In a third aspect, there is provided a system comprising a processor, a database comprising job requirements for at least one job position and at least one candidate profile data set, and a storage medium comprising instructions which when executed by the processor causes the processor to perform the method according to the first aspect.

20 In order that the present invention may be fully understood and readily put into practical effect, there shall now be described by way of non-limitative examples only preferred embodiments of the present invention, the description being with reference to the accompanying illustrative figures.

In the figures:

- 25 Figure 1 shows a schematic representation of an embodiment of the invention;
Figure 2 shows a schematic representation of an embodiment of the invention;
Figure 3 shows an example of skill clusters;
Figure 4 shows an example of a list of industries;
Figure 5 shows an example of a list of job functions;
30 Figure 6 shows an example of how the compatibility score is computed;
Figure 7 shows an example of how a location score is computed;
Figure 8A and 8B shows organisation and candidate preference forms respectively;

Figure 9 shows an example of two-sided matching;

Figure 10 shows a schematic flowchart of an embodiment of the invention;

Figure 11 shows an example of named-entity recognition (NER);

Figure 12 shows an example of a piecewise function of skill normalisation;

- 5 Figure 13 shows an example of a continuous piecewise linear function of the commute score against the commute time.

10 Figures 1 and 2 respectively show a schematic overview of an example of a workflow (or method) 100 and system 200 that allows candidates to be efficiently and optimally matched to job positions.

In Figure 1, at block 105 candidates looking for a job position including permanent, temporary and internships, login to a portal and filled in a candidate application form. The use of the portal and form allows for the relevant information to be provided by the candidates in the appropriate fields and makes the subsequent data processing easier. Alternatively, the candidate may provide a hard copy of their own curricula vitae which may be broken down by OCR methods into the respective fields. The candidate profile information may be referred to as the candidate's curriculum vitae (CV) herein in various forms be it a hard copy or entry via the portal described.

20

Companies looking to hire for a position provide a job posting, typically via a hiring manager, with a list of requirements of the job and expected remuneration and benefits. As an example, an online platform or portal may be provided that guides the hiring manager to fill in the requirements (both essential and preferred) in the respective fields in the job description application form in block 110, for example a job title, a description of the expected work, skills like language proficiency, proficiency in computer programming or specific programming language, work experience in specific areas like sales, or specific territories, educational requirements, et cetera. Alternatively, the hiring manager may provide a hard copy of the requirements which may be broken down by optical character recognition (OCR) methods into the respective fields.

30

Based on the candidate information and job description, a compatibility score and a capability score may be generated and combined into a final score as shown in block 115. In block 120, two way matching may be done to generate top n recommendations to each candidate. At block 120, candidates rank their given choices. A second two way match is done at block 130 which generates top n' (less than n) job assignments for the candidate. The candidate may subsequently go for interviews and placement as shown in blocks 135 and 140 respectively.

In an example, there are more job positions (jobs) than candidates. For example, there may be 50 candidates and 60 job positions.

Figure 2 show a system 200 that may be used in the scoring of a candidate with respect to a job position is performed. A skill score 215 is computed based on the similarity between the candidate profile 205 and the job profile 210. The compatibility score 220 is obtained from various individual scores, for example a job function score 230, a company type score 235, a salary fit score 240, an individual score 245, a location score 250, and an industry score 255. The scores may be modified based on the candidate preferences in block 26 to obtain the compatibility score 220. The company or organisation may have preferences as shown in block 265 that modifies the compatibility score 220 or final score 225. An example of the scores of a candidate is shown in block 285 – a skill score of 87, a compatibility score of 76 and a final score of 46. The calculation or computation of the scores are elaborated in greater detail below.

On the job posting, natural language processing (NLP) methods (like word embedding models) and machine learning are used to parse out skills and compare them against the skillset of candidates. For every skill in the job description, a comparison is made against the skillset of candidate and an aggregated score is generated.

A possible NLP method that may be used is named-entity recognition (NER). Every curriculum vitae (CV) or job description (JD) contains unstructured text and the format often varies largely across individuals or companies. Separating them into pre-defined categories is therefore a non-trivial task. For example, from the CV context, "Apple" could be a person's name but at the same time, "Apple" could also be a company. SpaCy, an industrial-

strength natural language processing library from Python may be used to process the raw text intelligently. It uses a word embedding strategy using subword features and a deep convolutional neural network with residual connections and a novel transition-based approach to named entity parsing. Figure 11 shows an example of how this works, the NER method recognises the company name, job title, period of employment, and employment summary from a candidate's CV.

Before the training process is performed, pre-processing and cleaning of the CVs (or resumes) (for example remove punctuation etc.) may be done. The model is trained with a huge pool of CVs, JDs and labelled entities. A possible list of entity annotations is provided below.

CV Entities	JD Entities
Full Name (FirstName, MiddleName, LastName)	Job Description
Email	Job Requirements
Gender	Skills (and calculated strength)
Contact Details	
Address (Address Line 1, Address Line 2, Postal Code)	
Country	
Executive Summary	
Education (Institution, Type, Field, Start Period, End Period, Duration, GPA)	
Employment (Company Name, Job Title, Start Period, End Period, Duration, Employment Summary)	
Skills (and calculated strength)	
Additional Links	
Activities (CCA/Hobbies)	
Leadership (Role, Summary, Duration)	
Volunteering (Activity, Duration)	

For some entities labelled above, it may be broken down into sub-entities using other methods. For example, for "Gender", a bi-directional long short-term memory (LSTM) model may be used to break the gender down into "Male" and "Female" based on a large corpus of data.

For skills, the position of where the skills appear in the resume matters as well (i.e. when the skills were learn or used). The strength of a skill extracted from recent employment is higher than a skill extracted from a job position several years before. A decaying function as shown below may be used to modify the strength of the skill with a time factor.

5 Decaying function: $Skill\ Strength = 100 - (\frac{x}{2})^3$

For example, if a candidate has indicated “Python” as a skill in his current employment, the skill strength would be 100.

- 10 Another NLP technique that may be used is word embedding. Using job descriptions, CVs and data from the web (for example text from Wikipedia), a model is trained to represent every skill as an n-dimension vector. If two skills are related to each other, (i.e. the two vectors are related to each other), and their corresponding vectors will be quite similar which will return a very high score (related skills) for such pairs. For example, “photoshop”
- 15 and “illustrator” has a high degree of similarity while “photoshop” and “CAD” has a low degree of similarity. Figure 3 shows an example of skill clusters. For example, for photo editing skills the knowledge of knowing how to use software like Photoshop, Sketch, Illustrator, and Acorn would require similar skills. For video editing skills, the knowledge of knowing how to use software like Sony Vegas, Adobe After Effects, Adobe Premiere Pro, and
- 20 Windows Movie Maker would require similar skills. For web development, knowledge of HTML and CSS would be considered similar skills compared to the photo editing or video editing skills.

Scoring Algorithm

- 25 The high level idea of the methodology is to harness 20 years of human resources analytics data and train a model that is able to translate mathematically the syntactic structures of phrases and sentences.

Capability Score and Compatibility Score

- 30 As shown in Figures 1 and 2, the capability score and compatibility score is computed based on each candidate’s CV relative to the job description. The capability score generally

comprises the skill score, course score, GPA score (or academic score), leadership score, and volunteer score. The compatibility score generally comprises the future skill (or skill/s to learn) score, industry score, job function score, location score, salary score, individual score, and psychometric score. Some or all of the components may be used for the capability score and compatibility score and depends on multiple factors including accuracy and companies' request and/or domain problem.

The NER model may be able to extract out most fields to a sufficient accuracy, but it is still a probabilistic output and may not be accurate under various circumstances. For example, the candidate may have provided an image of the skill instead of words in the CV, which does not allow for the image to be broken down by the NER model. This presents a problem, and the candidate may need to provide the information directly rather than the method extracting the information from the candidate's CV to provide the database. In cases, where the candidate does not or is not able to provide the missing information, the component of the score would be disregarded. In addition, the component is removed if we are not able to obtain a score better than a threshold. A simple logic may be used to check if the score is higher than the threshold. For example, if nothing is parsed from the CV the score is not able to meet the threshold score.

Skills score / Future skills score (skills to learn score)

Both "skills" and "skills to learn" score are calculated similarly. For "skills", the skills are extracted from the resume or CV of the candidates. The candidates may add/remove skills as well. For the training process, word embedding to calculate the pairwise score of two words may be used. It is mainly inspired from the paper, Efficient Estimation of Word Representations in Vector Space (Mikolov, Tomas; et al. (2013)), where the model takes input from a large "corpus of text" and produces a vector space by utilizing a continuous skip-gram neural network architecture. Herein, the large "corpus of text" means a large number of job descriptions, CVs and the English Wikipedia corpus, and every word is represented as a 300-dimension vector. A skill may also be more than a word. In most cases, we do a simple vector addition to all words. However, we also reduce the magnitude or filter words that appear more frequently among documents.

For every pairwise skill among candidate skills $\{s_j\}$ and job skills $\{s_k\}$, we measure the cosine similarity between the two vectors

$$s_{j1} \cdot s_{k1} = |s_{j1}| |s_{k1}| \cos \emptyset$$

- 5 At the same time, the strength of a candidate's skill $\{c_j\}$ may be accounted for by either requesting the candidates to rate their skills or predict their proficiency from the resume. The proficiency of the skills is normalised and may be focused on the relative strength among the skills of a candidate.
- 10 The skills parsed or mandated from the job description $\{s_k\}$ are then iterated against the candidate skillset $\{s_j\}$ and the skill that generated the highest score is recorded. The final skill score is then computed by taking the sum of every score along with a normalized requirement score $\{r_k\}$ predicted from the job description. If normalization is not done, the normalized requirement score $\{r_k\}$ may be removed. The normalization may be based on
- 15 the candidates' ranking or rating of their skills or a prediction of a candidate's proficiency from the candidate's CV and is elaborated further below. Thus, the normalization provides a candidate's relative standing with respect to the other candidates.

Equation 1:
$$\sum_{k=1}^n r_k \max_{j,k} c_j s_j s_k$$

- 20 Briefly, the method 1000 is shown in Figure 10 and comprises obtaining skills and proficiency from the candidate/s CV in block 1005, and obtaining skills and requirements from job descriptions (JDs) in block 1010. The data is pre-processed in block 1015 and the skills are converted into a vector through an embedding model in block 1020. In block 1025, the cosine similarity between pairwise skills is determined by the equation above. In block
- 25 1030, equation 1 is applied to all pairwise skills to generate a skill score.

The table below shows examples of various outputs from the skills embedding model.

Skill 1	Skill 2	Similarity (%)
Python	Java	65
Adobe Photoshop	Photoshop	100

Microsoft Word	MS Word	100
Microsoft Word	Office	79
Information Technology	Artificial Intelligence	57
Information Technology	Human Resources	12
Business Development	Sales	74
Microsoft	MNC	87

It is evident that the model is able to identify the same skill but written differently for example Microsoft Word and MS Word. The candidates may also be required to rate (or rank their skills) from a scale of 1 to 5, and is elaborated further below.

5

The ratings or rankings of the skills may be normalised. It is important to note that, the average skill rating will decrease overall with this normalization. Two possible methods to handle this normalization are provided as an example. The first method is to have a very general map for every skill rating as shown below.

Candidate rating	1	2	3	4	5
Normalised value	0.2	0.4	0.6	0.8	1

10

The problem with this method is that it relies on the candidate's truthful and accurate assessment of the candidate's own skill level, which may be less reliable and objective and difficult to apply consistently across all candidates.

15 In another method, based on all the candidates' rating or ranking of their skills, different percentiles of each skill is determined, for example the 25th percentile, the 50th percentile (or mean), and the 75th percentile. Other percentiles may also be used. The overall normalization of each candidate is also determined. Advantageously, this allows the differentiation between a candidate who indicates all '2's for a skill and another candidate
20 who indicates all '5's for a skill. For example, if a candidate indicates all 2s, overall normalization will be 0.7 and if a candidate indicated all 5s, overall normalization will be 1. Every percentile point is broken down and used to map a linear function as exemplified in

Figure 12. The end result is a piecewise function where every interval belongs to a certain percentile range. By using an empirical method, specific numbers are set to specific threshold and conditions.

- 5 In an example, a list of candidates provide the following ratings and the mapping from the method above is used.

Rating Distribution of Skills	Mapping
1,1,1	1→ 0.3
2,2,2	2→0.6
3,3,3	3→0.782
4,4,4	4→0.816
5,5,5	5→0.85
2,4	2→0.46, 4→0.83
2,5	2→0.45, 5→0.87
1,5	1→0.3, 5→0.874
1,1,1,1,1,4,5	1→0.3, 4→0.88, 5→0.968
4,4,5	4→0.681, 5→0.876

- 10 For “skills to learn”, the candidates provide skills that they wish to learn from or during their employment, and the skills to learn score may similarly be computed and ranked in accordance to their preference as the skills score. The skills score and skills to learn score are computed similarly as the former may be viewed as skills possessed by the candidate while the latter is the skills the candidate hope to possess. The skills to learn score may be particularly advantageous for candidates looking for internship positions, or at the initial stage of their working life (for example, new graduate or new to the industry).
- 15

Course (course-match) score

- For each candidate, the education (or courses attended) is extracted from the CV using the NER model described. Alternatively, the candidate may directly provide the information in the portal with an answer to a specific field or question. For every job posting, details of the specified courses needed and job title are obtained from the hiring company. If the courses specified by the employer is the same as the course from the candidate, there is a match otherwise there is not (i.e. MATCH = 1, NO MATCH = 0).
- 20

From the large corpus of employment data and CVs available, the following fields are extracted:

Education (from alma mater) – Field
First employment (from alma mater) – Job Title

For each education field (i.e. physics, computer science etc), the job functions of the people with a degree from a particular field are aggregated. For example, based from employment data, a sample breakdown of distribution from physics may be as shown in the table below:

Job Function	Percentage, CM
Research	32%
Education	18%
Engineer	15%
Data Science	12%
Higher Education	10%

Every major (or degree or specialization) is likely to have a different distribution of job functions.

The course-match score is calculated based on these aggregation data using the following function.

$$Course\ Match\ Score = \begin{cases} 80 + (MATCH * 20), & 30\% < CM \\ 80 + (MATCH * 20) - \left(\frac{CM-30}{6.45}\right)^3, & else \end{cases},$$

where CM is the percentage of the aggregated job functions for each major.

GPA score (academic results score or academic score)

The hiring company may also stipulate a minimum GPA, and may be formulated as a GPA score as shown below. Alternatively, the GPA may be in the form of the degree classification used in some countries. For example, the academic score may be a normal degree, a degree with honours, a degree with first class honours etc. Depending on the country, this may be translated into an equivalent GPA or the GPA score may be calculated from whether the degree classification meets the minimum requirement set.

$$GPA\ Score = \begin{cases} 0, & GPA_{min} < GPA \\ 100, & else \end{cases}$$

Leadership score

For every candidate, the NER method is used to extract out the role, summary and duration of the candidate's leadership position.

Based on employment data, each role (or position), summary and duration, may be categorised into 3 groups (director/captain roles typically belong to group A, vice/assistant director/captain roles typically belong to group B and leader/committee (i.e. treasurer etc) roles typically belong to group C). Every role and summary is sequenced, tokenized, and fed into a term frequency-inverse document frequency (tf-idf) model, which classifies them into the 3 groups respectively and may be scored as shown below.

$$Leadership\ Score = \begin{cases} 100, & group\ A \\ 75, & group\ B \\ 50, & group\ C \end{cases}$$

Volunteer score

For each candidate, the NER model is used to extract any volunteer activity and duration. The total volunteering duration V_{sum} is obtained by the addition of all the activity and duration. The company may specify a minimum required voluntary duration V_{min} for the job position and the volunteer score computed as shown below. This is particularly advantageous for candidates with no or little prior work experience which may make it difficult to distinguish between candidates.

$$Volunteer\ Score = \begin{cases} 100, & V_{sum} > V_{min} \\ 0, & else \end{cases}$$

Industry/job function score

The NER model is used to extract out the industries and job functions from both the candidate CV and the job requirements. The candidates and companies may edit and rank the fields as well.

A candidate may be given the option to rank their top 5 choices for industry preferences in a form (online or offline), based on which a score between 0 and 100 will be generated. In an example, a pre-defined list of 30 industries is shown in Figure 4. This is analogous to the candidates rating / ranking their individual skill but the rank is calculated differently here.

For every rank R_i , we allocate a rank to a weight W_i . The weights are calculated as follows:

$$W_i = (N - R_i + 1) * p,$$

where p is an exponent parameter to control distribution of the weights.

A candidate may be provided an option to pick a maximum of 2 preferred job functions in a form (online or offline), based on which a score between 0 and 100 will be generated. In an example, a pre-defined list of 16 job functions is shown in Figure 5.

In another example, some possible ways to differentiate different job functions (or job types) include:

A) Front Office/Back Office prediction for CVs and JDs (i.e. Front Office vs Back Office); and

B) Nature of job scope for CVs/JDs (i.e. Structured/Well-defined vs Independent/Unstructured).

A front office job may typically be a job that requires direct interaction or communication with people in particular clients and customers, whereas a back office job is more towards behind the scenes, where the candidate provides essential operations to the company. Some front office jobs include sales and hospitality roles while back office jobs include accounting, software engineering and graphic design. However, it is also important to note that these roles may also be a hybrid of both, i.e. a combination of both back office and front office (product manager, data analyst roles). A structured/well-defined job is more task oriented, where job roles are highly well-defined while an independent/unstructured job is highly independent and requires little supervision.

Subsequently, the word embedding model is used to calculate the cross-similarity across industries and job functions. For example, if a candidate indicated “Business Development” as a top job function preference and the job has a “Sales” job function. This may be similarly applied to find similarities between the candidate’s preferred industries and the industry of the job position. The score will be calculated as S_i , 74. The final score is mapped as:

$$JobFunction\ Score / Industry\ Score = \begin{cases} 0, & S_i < 50 \\ \min(\sum_I^N W_i * S_i, 100), & S_i \geq 50 \end{cases}$$

All the values are summed up across candidate rankings for each of the Job Function score and the Industry Score, and each of the final score is capped at 100.

Company Type Score

- 5 Candidates may also have a preference towards working in certain companies for various reasons, be it for interest or exposure. This may be taken into account as well to provide a better fit between the company and candidate. In an example, candidates select their top 3 choices from the following list:
- 1) Government
 - 10 2) Multinational Corporation (MNC)
 - 3) Small & Medium Enterprises (SME)
 - 4) Non-profit Organisations
 - 5) Start-up
 - 6) Home Based Office/Remote
- 15 A score will be generated for every job.

In an example, by using various psychometric features, front office/back office and nature of job scope information, the preferential probability for each candidate for each company type is calculated and a score is generated for every job.

20

Location score

- For every job and candidate, a commuting time (based on public transport) may be calculated. Generally speaking, the lower the commute time, higher will be the location score. Using a corpus of employment data, every commute time is plotted into a distribution
- 25 and selected percentile points are broken into intervals. For every interval period, a simple linear regression may be used to map every single commute time into a score. The final result is a continuous piecewise linear function as shown in Figure 13. In this example, a candidate with a commute time of 10 mins or less (before 15th percentile) will have a location score of 100.

30

It is important to note that the percentile may vary for every job function and the distribution percentile points are broken down for every job function. This may be

attributed to certain job functions and/or industries being located in specific areas that may skew the data.

Salary Score

- 5 Each candidate may provide the expected salary (C_S) for a job position. The company may provide a salary range J_{min} and J_{max} (It is possible that J_{min} and J_{max} are the same). A piecewise function is mapped from the information provided as follows

$$Salary\ Score = \begin{cases} \max(0, 100 - \left| \frac{C_S - J_{min}}{C_S} * 100 \right|), & C_S < J_{min} \\ 100, & J_{min} \leq C_S < J_{max} \\ \max(0, 100 - \left| \frac{C_S - J_{max}}{C_S} * 100 \right|), & J_{max} < C_S \end{cases}$$

10 *Psychometric score*

A psychometric score for each candidate may be obtained via a test administered via the portal or via an external provider which subsequently provides the score to the candidate's profile information in the database.

15 *Individual Score*

Candidate will fill in their preferences in a form and hiring managers will also fill in a job posting. As an example, candidates are asked certain questions (see Figure 8B) and their individual choices are used to compare their compatibility against the respective jobs.

- 20 In an example, the individual score may be derived from the same parameters as the job function and may compose of:

A) Front Office/Back Office prediction for CVs and JDs (i.e. Front Office vs Back Office) (Front Office Score); and

B) Nature of job scope for CVs/JDs (i.e. Structured/Well-defined vs
25 Independent/Unstructured) (Job Scope Score).

Each candidate and hiring manager for the job would select a score from 0 to 100 for each of the Front Office Score and Job Scope Score, this can be by inputting numerically or by moving a slider.

- 5 The individual score is calculated by the following:

$$\begin{aligned} & \text{Individual Score (Front Office Score or Job Scope Score)} \\ &= 100 - |S_{\text{candidate}} - S_{\text{employer}}| \end{aligned}$$

- Thus, the individual score may be computed based on a numerical difference between the candidate's preferred job function and the hiring company's categorisation of the job function. The individual score reflects numerically the extent of the match between the candidate's preferences and the actual job function as provided by the company. When there is an identical match between the job function preference of the candidate and that as set by the company, the difference is zero and is reflected in a maximum individual score of 100. The individual score may be computed as an average of the Front Office score and Job Scope score, or by using either the Front Office Score or Job Scope score. The average of the Front Office score and Job Scope score may be viewed as taking both scores equally, but it is also possible for the Individual Score to be made up of the Front Office score and Job Scope score in different weightages based on the hiring company's selection.

Organisational Metrics

- 20 Companies may value the components of each score differently based on the different organisational values. The companies have the flexibility to increase or reduce the importance of a component, or completely remove a component which they do not require to best meet the companies' requirements.
- 25 For example, if a client sees only skills, courses and leadership as valid, the capability may be made up of the skill score, course score, and leadership score. I.e. Capability score = Skill Score + Course Score + Leadership Score

Certain domains may also have specific requirements. For example, technical jobs (like a software programmer) may have a greater emphasis on the relevant technical skills required for the job.

5 *Capability Score*

Every organisation has a different structure and values metrics differently. The companies have the flexibility to increase or reduce the importance of a component, or completely remove a component to best meet the companies' requirements. As mentioned above, the capability score may be composed of three components to the score: skill, leadership and course. However, each component may not be equal. The hiring company may indicate for every component their emphasis on it, for example a value from 0 to 1. Figure 8A shows an example of a possible interface to allow a hiring manager to set the company's preference. alternatively, the hiring manager may enter the value directly.

- 15 For example, if a client sees only skills, courses and leadership as valid, the capability may be made up of the skill score, course score, and leadership score. I.e. Capability score = Skill Score + Course Score + Leadership Score

20 Certain domains may also have specific requirements. For example, technical jobs (like a software programmer) may have a greater emphasis on the relevant technical skills.

For example, the client may value "Skill" t_s as 0.5, "Course" t_c as 0.5 and "Leadership" t_L as 0.9. t_i represents the weight of every component with a value from 0 to 1, and n is the total number of components.

$$25 \quad \text{Max Score of Skill} = \frac{t_s}{t_s + t_c + t_L} * (100 - 15n) + 15$$

$$\text{Max Score of Course} = \frac{t_c}{t_s + t_c + t_L} * (100 - 15n) + 15$$

$$\text{Max Score of Leadership} = \frac{t_L}{t_s + t_c + t_L} * (100 - 15n) + 15$$

In this example, the max possible skill score is 29.5, the max possible course score is 29.5, the max possible leadership score is 41, and the total possible capability score is 100.

Compatibility score

- 5 Compatibility takes into account candidates' motivation and interests entering the workforce programme. Through a carefully designed form, the candidates' personal preference are accounted for. Some preferences include job function, industry preference, location, individual preferences, and company type preferences. Based on these preferences, weighted score is calculated for the candidates for each job.

10

For the compatibility score, each candidate values a metric differently differently (i.e. a candidate might prefer employment nearby their residence rather than employment far away) and an arbitrary function that is suitable for every candidate is inaccurate.

- 15 The candidates are required to rank their priorities for a job. In an example, compatibility score is composed of the location score, the job function score, the company type score, the salary score, the skills to learn score and the individual score. The candidate ranks these six metrics in preference. Figure 6 shows an example of how this may be done.

- 20 Each ranking may be annotated as $R_{location}$, $R_{jobfunction}$, $R_{companytype}$, R_{salary} , $R_{skillstolearn}$, $R_{individual}$ and their scores as $S_{location}$, $S_{jobfunction}$, $S_{companytype}$, S_{salary} , $S_{skillstolearn}$, $S_{individual}$ respectively.

For every rank R_i , a weight W_i is allocated and ranked by the rank exponent as follows:

25

$$W_i = (N - R_i + 1) * p ,$$

where p is an exponent parameter to control distribution of the weights and N is the total number of components available.

30

Compatibility is calculated to be:

$$\begin{aligned} \text{Compatibility Score} = & 0.2 * (S_{jobfunction}) + \\ & 0.8 * [(W_{jobfunction} * S_{jobfunction}) + \\ & (W_{location} * S_{location}) + \\ & (W_{companytype} * S_{companytype}) + \end{aligned}$$

$$(W_{salary} * S_{salary}) + (W_{skillstolearn} * S_{skillstolearn}) + (W_{individual} * S_{individual})]$$

For example, if a candidate indicates $R_{location}$, $R_{jobfunction}$, $R_{companytype}$, R_{salary} , $R_{skillstolearn}$, $R_{individual}$ to be 1, 2, 3, 4, 5, 6 respectively, the max scores for each component will be 23, 39, 15.2, 11.4, 7.6, and 3.8 respectively.

5

In another example, the company score is not used in the compatibility score, for example if the company score is not sufficiently accurate.

The formula would be as follows:

$$\begin{aligned} \text{Compatibility Score} = & 0.2 * (S_{jobfunction}) + \\ & 0.8 * [(W_{jobfunction} * S_{jobfunction}) + \\ & (W_{location} * S_{location}) + \\ & (W_{salary} * S_{salary}) + \\ & (W_{skillstolearn} * S_{skillstolearn}) + (W_{individual} * S_{individual})] \end{aligned}$$

10 Two-sided matching algorithm

The two-sided matching is solved using a bipartite graph with candidates and jobs as the two groups to perform matching on. After the compatibility and capability scores are calculated, the matching is done in 2 stages.

15 The following are some assumptions that are made:

1. Candidates are shown a list of jobs which is equal to n_i
2. Each company will see m_j candidates
3. Companies cannot expect to have all their job positions filled

20 Assuming that there are n number of candidates and m number of jobs, in matrix notation the scores for each candidate against each job can be represented by

$$\{u_{ij}\}_{n,m}$$

where u_{ij} is the score (based on compatibility and relevance) for i^{th} candidate against j^{th} job.

The matrix is of dimension $(m \times n)$.

25

These scores are converted into a rank matrix which is represented as

$$\{r_{ij}\}_{n,m}$$

where r_{ij} represents the rank of i^{th} candidate against j^{th} job. Basically, the jobs are ranked for each of the candidates.

5 The objective function is given by:

$$\min_{i,j} \sum \sum r_{ij} \cdot x_{ij}$$

where the constraints are:

$$x_{ij} \in \{0,1\}$$

$$\sum x_{ij} = n_i, \forall i$$

$$\sum x_{ij} = m_j, \forall j$$

$$\sum n_i = \sum m_j$$

- 10 This problem is solved using a min-cost flow algorithm, and preferably by the network simplex method. Figure 9 shows a brief overview of the min-cost flow problem. To start, the problem parameters are converted to a min-cost flow network. A min-cost flow network consists of nodes that have a parameter "demand" associated with it. The edges of a min-cost flow network have "capacity" and "cost". The candidates and the jobs form the nodes.
- 15 The job nodes are treated as demand nodes and the candidates are treated as supply nodes. As a consequence, job impressions m_j is a positive demand parameter and the candidate impressions n_i is negative demand parameter. Every pair of candidates and jobs are connected by a unit capacity edge with the cost given by r_{ij} . The theory of min-cost flow shows that this formulation is equivalent to the integer linear programming (ILP) construct
- 20 given above. The final optimal flow and x_{ij} in the ILP above coincide. Only the x_{ij} which contributes to rank r_{ij} will be shown as a job recommendation to the candidate and a candidate recommendation to a company.

- In contrast with the max flow algorithm, the flow is fixed for the min flow i.e. total job impressions from the source and the total number of candidate impressions to the destination is fixed beforehand. The edges from the source to the candidates have a capacity of n_i and the cost per edge is 0, i.e. no restrictions. The internal edges have a
- 25

capacity of x_{ij} and the edges from the job nodes to the destination have a capacity of m_j . Only the x_{ij} which contributes to rank r_{ij} will be shown as a job recommendation to the candidate and a candidate recommendation to a company.

- 5 To illustrate using an example as shown in Figure 9, there are n ($= 50$ in Figure 9) candidates and each candidate is given a list of 5 jobs to choose. Hence, there is a flow of 5 from the source to candidate network. The connections between the candidates and the jobs have a weight derived from a ranking based on the score of each candidate against each of the available jobs. This is a combination of compatibility and capability scores for each
10 candidate. For example, candidate 50 was ranked 200 for job 200. The weights on the job to target network can have a maximum value of 5 since each company can view a maximum of 5 candidates for each job.

- Using min cost flow, this problem is solved in which optimal weights in the job to target
15 network are assigned as well as recommendations for top n jobs for each candidate by having a sparse network. These optimal weights will determine the number of candidates received by the company.

- To illustrate using an example, there are n candidates and each candidate is given a list of 5
20 jobs to choose from. Hence, we see the flow of $5n$ in the min-cost flow network. The connections between the candidates and the jobs have weight derived from a ranking based on the score of each candidate against each of the available jobs. This score is a combination of compatibility and capability scores for each candidate.

- 25 In the above scenario, r_{ij} is the average of the compatibility and capability scores. Capability scores take into account the candidate perspective and the capability scores take into account the company perspective. But, taking an average might not yield the best allocation and hence may be further optimized.

- 30 The original matching process may be broken down into two parts by running the min-cost flow algorithm twice: once from the job perspective (capability) and once from the candidate perspective (compatibility).

This may be achieved by dividing the number of candidate impressions by 2 and the result (rounded to nearest integer) is fed to the min-cost flow network. r_{ij} is the capability score in this case. The company sees the appropriate number of candidates that is set (for example
5 five candidates). In the second round, the algorithm is run again with r_{ij} as the compatibility score and the remaining candidate impressions are allocated to a job. In the scenario where a candidate and company both like each other, conflicts can arise when the same candidate is allocated to a job in both rounds. This conflict may be resolved by removing the edges that were matched in the first round before running the second round of allocation.

10

Each stage has 2 inputs and 1 output.

In Stage 1:

- 1) Inclusion matrix ($n * m$) consisting of all ones
- 2) Rank matrix ($n * m$) where the ranking can be 1 to m for each n

15

This ranking may be derived either using the final score or based on compatibility and capability score individually. The latter may be viewed as 2 rank matrices, one based on compatibility and one based on capability

- 3) Output is a recommendation matrix ($n * m$). This is a matrix of 0 and 1 where 1 indicates the jobs that have been assigned to each candidates and 0 means no
20 match.

20

Stage2

- 1) Inclusion matrix ($n * m$) here is the recommendation matrix from stage 1
- 2) Rank matrix is generated from candidate ranking to jobs recommended from
stage 1
- 3) Output matrix is matrix of 0 and 1 where 1 indicates the jobs that have been
25 assigned to each candidates and 0 means no match.

25

Thus, in Stage 1 the capability score and/or compatibility score is used in the matching algorithm to generate a first set of job recommendations for each candidate. In Stage 2, the
30 matching algorithm may be run again with each candidate's ranking of each job recommendation in the first set of job recommendations to generate a second set of job

30

recommendations. Stage 2 accounts for the candidate's preferences for the job recommendations to provide an optimum fit between candidates and job positions.

In an example, by running the capability score and compatibility score individually, each candidate may be provided with 10 recommendations – 5 recommendations each from running the matching algorithm with the capability score and the compatibility score respectively. In other words, each candidate receives 5 matches ("1") based on the compatibility score and 5 matches based on the capability score. It will be apparent that there is no limitation to the number of recommendations per se and may be set accordingly.

In an example of the matching algorithm, the following tables may be generated.

Table 1

	Job A	Job B	Job C
Candidate A	66	55	21
Candidate B	65	55	77
Candidate C	77	6	45

Table 1 shows the scores (based on the average of the compatibility and capability scores) generated for each candidate for the available jobs in the database.

Table 2

	Job A	Job B	Job C
Candidate A	66	<u>55</u>	21
Candidate B	65	<u>55</u>	<u>77</u>
Candidate C	<u>77</u>	6	<u>45</u>

Table 2 shows the results of a run of the allocation (or matching) algorithm with the bold and underlined scores indicating that the candidate is matched to the job. From Table 2, it may be observed that although Candidate B has a higher score for Job A compared to Job B, Candidate B is not recommended job A as Job A is too popular and there are candidates more suitable for Job A. It may be possible for a manual match to be made and recommend

a candidate manually despite the system and method not doing so. For example, Candidate A may be recommended for Job C, after which the allocation algorithm may be run again to redistribute the remaining jobs and candidates. This may be particularly advantageous in educational institutes where manual matches may be required.

5

The generated job recommendations may be provided to each candidate to allow each candidate to rank their preference as shown in block 120 in Figure 1 and stage 2 in the matching method above. A second two way matching is performed again to generate top job recommendations (which is less than the first set of job recommendations ranked by the candidate) to the candidate as shown in block 130. The candidate may proceed to interview with the company as shown in block 135 and placement in block 140.

10

As an example, the candidate may rank the first set of five job recommendations from 1 to 5, and subsequently only the top 2 choices in the second set of job recommendations are sent to the company.

15

The methods and systems described herein allows for the optimal allocation of candidate/s to job position/s and reduces or avoids bias in the allocation.

Any element expressed herein as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a combination of elements that performs that function. Furthermore the invention, as may be defined by such means-plus-function claims, resides in the fact that the functionalities provided by the various recited means are combined and brought together in a manner as defined by the appended claims. Therefore, any means that can provide such functionalities may be considered equivalents to the means shown herein.

20

25

In various embodiments, modules or software can be used to practice certain aspects of the invention. For example, software-as-a-service (SaaS) models or application service provider (ASP) models may be employed as software application delivery models to communicate software applications to clients or other users. Such software applications can be downloaded through an Internet connection, for example, and operated either

30

independently (e.g., downloaded to a laptop or desktop computer system) or through a third-party service provider (e.g., accessed through a third-party web site). In addition, cloud computing techniques may be employed in connection with various embodiments of the invention. In certain embodiments, a “module” may include software, firmware, hardware,
5 or any reasonable combination thereof.

Moreover, the processes associated with the present embodiments may be executed by programmable equipment, such as computers. Software or other sets of instructions that may be employed to cause programmable equipment to execute the processes may be
10 stored in any storage device, such as a computer system (non-volatile) memory. Furthermore, some of the processes may be programmed when the computer system is manufactured or via a computer-readable memory storage medium.

It can also be appreciated that certain process aspects described herein may be performed
15 using instructions stored on a computer-readable memory medium or media that direct a computer or computer system to perform process steps. A computer-readable medium may include, for example, memory devices such as diskettes, compact discs of both read-only and read/write varieties, optical disk drives, and hard disk drives. A computer-readable medium may also include memory storage that may be physical, virtual, permanent,
20 temporary, semi-permanent and/or semi-temporary.

A “computer,” “computer system,” “computing apparatus,” “component,” or “computer processor” may be, for example and without limitation, a processor, microcomputer, minicomputer, server, mainframe, laptop, personal data assistant (PDA), wireless e-mail
25 device, smart phone, mobile phone, electronic tablet, cellular phone, pager, processor, fax machine, scanner, or any other programmable device or computer apparatus configured to transmit, process, and/or receive data. Computer systems and computer-based devices disclosed herein may include memory for storing certain software applications used in obtaining, processing, and communicating information. It can be appreciated that such
30 memory may be internal or external with respect to operation of the disclosed embodiments. The memory may also include any means for storing software, including a hard disk, an optical disk, floppy disk, ROM (read only memory), RAM (random access

memory), PROM (programmable ROM), EEPROM (electrically erasable PROM) and/or other computer-readable memory media. In various embodiments, a “host,” “engine,” “loader,” “filter,” “platform,” or “component” may include various computers or computer systems, or may include a reasonable combination of software, firmware, and/or hardware.

5

In various embodiments of the present invention, a single component may be replaced by multiple components, and multiple components may be replaced by a single component, to perform a given function or functions. Except where such substitution would not be operative to practice embodiments of the present invention, such substitution is within the scope of the present invention. Any of the servers (or database) described herein, for example, may be replaced by a “server farm” or other grouping of networked servers (e.g., a group of server blades) that are located and configured for cooperative functions. It can be appreciated that a server farm may serve to distribute workload between/among individual components of the farm and may expedite computing processes by harnessing the collective and cooperative power of multiple servers. Such server farms may employ load-balancing software that accomplishes tasks such as, for example, tracking demand for processing power from different machines, prioritizing and scheduling tasks based on network demand, and/or providing backup contingency in the event of component failure or reduction in operability.

20

In general, it will be apparent to one of ordinary skill in the art that various embodiments described herein, or components or parts thereof, may be implemented in many different embodiments of software, firmware, and/or hardware, or modules thereof. The software code or specialized control hardware used to implement some of the present embodiments is not limiting of the present invention. For example, the embodiments described hereinabove may be implemented in computer software using any suitable computer programming language such as .NET, SQL, MySQL, or HTML using, for example, conventional or object-oriented techniques. Programming languages for computer software and other computer-implemented instructions may be translated into machine language by a compiler or an assembler before execution and/or may be translated directly at run time by an interpreter. Examples of assembly languages include ARM, MIPS, and x86; examples of high level languages include Ada, BASIC, C, C++, C#, COBOL, Fortran, Java, Lisp, Pascal, Object

30

Pascal; and examples of scripting languages include Bourne script, JavaScript, Python, Ruby, PHP, and Perl. Various embodiments may be employed in a Lotus Notes environment, for example. Such software may be stored on any type of suitable computer-readable medium or media such as, for example, a magnetic or optical storage medium. Thus, the operation and behavior of the embodiments are described without specific reference to the actual software code or specialized hardware components. The absence of such specific references is feasible because it is clearly understood that artisans of ordinary skill would be able to design software and control hardware to implement the embodiments of the present invention based on the description herein with only a reasonable effort and without undue experimentation.

Various embodiments of the systems and methods described herein may employ one or more electronic computer networks to promote communication among different components, transfer data, or to share resources and information. Such computer networks can be classified according to the hardware and software technology that is used to interconnect the devices in the network, such as optical fiber, Ethernet, wireless LAN, HomePNA, power line communication or G.hn. The computer networks may also be embodied as one or more of the following types of networks: local area network (LAN); metropolitan area network (MAN); wide area network (WAN); virtual private network (VPN); storage area network (SAN); or global area network (GAN), among other network varieties.

The computer network may be characterized based on functional relationships among the elements or components of the network, such as active networking, client-server, or peer-to-peer functional architecture. The computer network may be classified according to network topology, such as bus network, star network, ring network, mesh network, star-bus network, or hierarchical topology network, for example. The computer network may also be classified based on the method employed for data communication, such as digital and analog networks.

As employed herein, an application server may be a server that hosts an API to expose business logic and business processes for use by other applications. Examples of application

servers include J2EE or Java EE 5 application servers including WebSphere Application Server. Other examples include WebSphere Application Server Community Edition (IBM), Sybase Enterprise Application Server (Sybase Inc), WebLogic Server (BEA), JBoss (Red Hat), JRun (Adobe Systems), Apache Geronimo (Apache Software Foundation), Oracle OC4J (Oracle Corporation), Sun Java System Application Server (Sun Microsystems), and SAP Netweaver AS (ABAP/Java). Also, application servers may be provided in accordance with the .NET framework, including the Windows Communication Foundation, .NET Remoting, ADO.NET, and ASP.NET among several other components. For example, a Java Server Page (JSP) is a servlet that executes in a web container which is functionally equivalent to CGI scripts. JSPs can be used to create HTML pages by embedding references to the server logic within the page. The application servers may mainly serve web-based applications, while other servers can perform as session initiation protocol servers, for instance, or work with telephony networks. Specifications for enterprise application integration and service-oriented architecture can be designed to connect many different computer network elements. Such specifications include Business Application Programming Interface, Web Services Interoperability, and Java EE Connector Architecture.

Embodiments of the methods and systems described herein may divide functions between separate CPUs, creating a multiprocessing configuration. For example, multiprocessor and multi-core (multiple CPUs on a single integrated circuit) computer systems with co-processing capabilities may be employed. Also, multitasking may be employed as a computer processing technique to handle simultaneous execution of multiple computer programs.

In various embodiments, the computer systems, data storage media, or modules described herein may be configured and/or programmed to include one or more of the above-described electronic, computer-based elements and components, or computer architecture. In addition, these elements and components may be particularly configured to execute the various rules, algorithms, programs, processes, and method steps described herein.

Whilst there has been described in the foregoing description preferred embodiments of the present invention, it will be understood by those skilled in the technology concerned that

many variations or modifications in details of design or construction may be made without departing from the present invention.

CLAIMS

1. A computer-implemented method for matching a potential candidate desirous in seeking employment to a job position, the method comprising:
 - 5 (a) providing a database comprising at least 50 job positions each with a job description, job requirements for at least 50 job positions, and at least 50 candidate profile data sets, wherein the job requirements are provided by analysing said job description by natural language processing methods to identify the job requirements;
 - 10 (b) computing a job score for a candidate for each job position by comparing the candidate profile data set with said job requirements; and
 - (c) generating a set of job recommendations for each candidate based on the job score by a minimum cost flow method.
- 15 2. The method according to claim 1, wherein the job description is analysed by parsing the job description and matching the parsed data to a set of abilities data to identify job requirements associated with the job position.
- 20 3. The method according to any one of claims 1 to 2, wherein the database comprises at least one candidate and the method comprises creating the candidate profile data set associated with said candidate.
- 25 4. The method according to any one of claims 1 to 3, wherein the candidate profile data set comprises at least one selected from the group comprising of skills competency, skills to learn, courses taken, academic results, leadership qualities, involvement in volunteer work, the candidate's preferred industry, the candidate's preferred job function, the candidate's location, the candidate's expected salary, the candidate's company type preference and psychometric data of the candidate, and computing a skills score, a skills to learn score, a course score, an academic results score, a leadership score, a volunteer score, an industry score, a job function score, a location score, a salary score, a company
- 30

type score, and a psychometric score respectively based on the available data in the candidate profile data set and an individual score based on the candidate's preferred job function.

- 5 5. The method according to any one of claims 1 to 4 wherein the job score is based on an average of a candidate capability score and a candidate compatibility score, or by using each of the candidate capability score and the candidate compatibility score as the job score to generate the set of job recommendations.
- 10 6. The method according to claim 5, wherein the candidate capability score is computed based on the skills score, the course score, and the leadership score.
- 15 7. The method according to claim 6, wherein the candidate capability score is further computed based on at least one of the academic results score and the volunteer score.
- 20 8. The method according to any one of claims 6 to 7, wherein the candidate capability score is computed based on weightages in the job requirements.
- 25 9. The method according to any one of claims 6 to 8, wherein the skills score is computed by iterating each skill in the job requirements against the skills competency in the candidate profile data set; recording a highest score generated for each skill; summing the highest scores for each skill in the job requirements to generate the skills score.
- 30 10. The method according to claim 9, wherein the skill score is normalised based on a ranking or rating of the candidate's skills in the skills competency data.
- 30 11. The method according to claim 10, wherein the skill score is normalised by combining the ranking or rating of each skill of all the candidates to generate a distribution curve; selecting multiple percentile points to map a function; and

generating a normalised requirement score for each skill and used in summing the highest scores to generate the skills score.

- 5 12. The method according to any one of claims 5 to 11, wherein at least one of the following conditions is fulfilled:
- (a) the course score is computed by determining if there is a first match between a course taken by the candidate and a course requirement in the job requirements; and computing the course score based on whether the first match is present and an existing employment data set;
- 10 (b) the academic results score is computed by determining if the academic results in the candidate profile data set meet a minimum threshold in the job requirements;
- (c) the leadership score is computed by classifying the leadership qualities in the candidate profile data set into one of a plurality of groups, each group having a specific value and classified based on an existing employment data set; and
- 15 (d) the volunteer score is determined by determining if the involvement in volunteer work in the candidate profile data set meet a minimum threshold in the job requirements.
- 20 13. The method according to any one of claims 5 to 12, wherein the compatibility score is computed based on at least one of the following components: the skills to learn score, the industry score, the job function score, the location score, the salary score, the individual score and the psychometric score.
- 25 14. The method according to claim 13, wherein the components of the compatibility score may be weighted according to the job requirements.
15. The method according to claim 13, wherein at least one of the following conditions is fulfilled:
- 30 (a) the skills to learn score is computed by iterating each skill in the job requirements against the skills competency in the candidate profile data set;

- recording a highest score generated for each skill; summing the highest scores for each skill in the job requirements to generate the skills score;
- (b) the industry score is computed based on a similarity between the candidate's preferred industry and an industry of the job position, and a rank in the candidate's preferred industry;
- (c) the job function score is computed based on a similarity between the candidate's preferred job function and the job function in the job requirements, and a rank in the candidate's preferred job function;
- (d) the location score is computed by determining a commute time between the candidate's location and a location of the job position; and determining the location score based on a function derived from a plotted distribution of commute time for the job position or similar job positions;
- (e) the salary score is computed based on an extent the candidate's expected salary falls within a salary range in the job requirements;
- (f) the individual score is computed based on the difference between the candidate's preferred job function and a job function in the job requirements.
16. The method according to any one of claims 1 to 15, wherein a network simplex method is used in generating the set of job recommendations.
17. The method according to any one of claims 1 to 16, further comprising generating a second set of job recommendations for each candidate based on a ranking of the candidate's preferences.
18. The method according to any one of claims 1 to 17, wherein there are more job positions than candidates.
19. A non-transitory medium comprising instructions which when executed by a processor causes the processor to perform the method according to any one of claims 1 to 18.