On the Exploitation of Textual Descriptions for a Better-informed Task Assignment Process

Nikos Kanakaris[a], Nikos Karacapilidis[b] and Georgios Kournetas[c]
Industrial Management and Information Systems Lab, MEAD, University of Patras, 26504 Rio Patras, Greece

Keywords: Natural Language Processing, Machine Learning, Operations Research, Human Resource Allocation.

Abstract: Human resources always play a crucial role on firms' profitability and sustainability. For a long time already, the identification of the appropriate personnel possessing the skills and expertise required to undertake a task is mainly performed through simple keyword searches exploiting structured data. In this paper, we build on Natural Language Processing techniques to take this identification to a higher level of detail, granularity and accuracy. Our approach elaborates unstructured data such as descriptions and titles of tasks to extract valuable information about the employees' capability of successfully engaging with a particular task. Text classifiers such as naive Bayes, logistic regression, support vector machines, k-nearest neighbors and neural networks are being tested and comparably assessed.

1 INTRODUCTION

Admittedly, the success or failure of an organization relies significantly on how its human resources are selected and assigned to its pending tasks (Arias et al., 2018). Despite the fact that both the personnel selection process and the proper assignment of personnel to an organization's pending tasks are of vital importance, these are mostly carried out today through ad-hoc techniques, which in turn rely on the managers' subjective (and sometimes distorted) opinion on how to allocate tasks to employees. In the majority of cases, managers decide using their tacit knowledge, which is frequently biased in favor or against certain employees (Sullivan et al. 1988). Hence, the exploitation of the skills and competences of the employees is not always maximized, which certainly affects the overall productivity of the organization.

Aiming to maximize the utilization of human resources, various AI-based approaches have been already proposed in the literature. These approaches assist in selecting and assigning employees to tasks. Most of them process structured data to maximize/minimize diverse Key Performance Indicators (KPIs). For a long time already, the identification of the appropriate personnel possessing the skills and expertise required to undertake a task is mainly performed through simple (Boolean) keyword searches exploiting structured data. This makes the above approaches incapable of tackling real-world problems where, in most cases, only unstructured textual data is available.

Elaborating previous work on the synergy between Machine Learning (ML) and Operations Research tools (Kanakaris et al., 2019; Kanakaris et al., 2020), this paper proposes a new approach that utilizes unstructured textual data to assign employees to tasks through a more detailed, granular and accurate task assignment process. The proposed approach is divided into two phases:

- **Personnel Selection:** the estimation of how relevant/qualified each employee is to undertake each task;
- **Human Resource Allocation:** the assignment of employees to tasks in a way that the total relevance is maximized.

The remainder of the paper is organized as follows: Section 2 discusses related work considered in the context of our approach, which is analytically described in Section 3. A set of experiments using

[a] https://orcid.org/0000-0001-9352-5807
[b] https://orcid.org/0000-0002-6581-6831
[c] https://orcid.org/0000-0001-8668-296X

Kanakaris, N., Karacapilidis, N. and Kournetas, G.
On the Exploitation of Textual Descriptions for a Better-informed Task Assignment Process.
DOI: 10.5220/0009151603040310
In *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems (ICORES 2020)*, pages 304-310
ISBN: 978-989-758-396-4

various ML classifiers is described in Section 4. Finally, limitations, future work directions and concluding remarks are outlined in Section 5.

2 RELATED WORK

2.1 Personnel Selection

A set of existing personnel selection approaches builds on concepts and techniques from the area of recommender systems. For instance, (Alkhraisat, 2016) proposes a new architecture for automated issue tracking system that is based on ontology, semantic similarity measures and document similarity techniques. This approach searches for similar issues and recommends candidate experts and related material. Adopting a similar research direction, (Heyn and Paschke, 2013) describes an extension for the Jira platform, which, for a given issue, recommends experts and related material. The benefits of this extension are the reuse of similar work and the distribution of the work to the correct developers. Another expertise recommender engine, which leverages knowledge uncovered from the profile of each user, can be found in (McDonald and Ackerman, 2000).

A different set of approaches to the personnel selection process builds on the utilization of appropriate Machine Learning algorithms. For instance, (Azzini et al., 2018) describes the application of various ML classifiers to extract knowledge about academic candidates from a semi-structured interview in the form of a questionnaire. Their main goal was to uncover the soft skills of individuals and match these skills with job requirements. Their empirical results demonstrated that Support Vector Machine (SVM) and naive Bayes approaches were more efficient than Decision Trees, k-Nearest Neighbors (k-NN) and Random Forests.

Another application of ML classifiers can be found in (Wowczko, 2015). The input data of this work were job titles and job descriptions, as they were published in job advertisements in Ireland in 2014. The proposed solution results in a categorization of jobs, eventually inferring which skills are affiliated with each job categorization. In this work, naive Bayes and k-NN algorithms were distinguished from many classification algorithms for their accuracy levels.

2.2 Human Resource Allocation

From an Operations Research (OR) perspective, a series of techniques and tools have been proposed and extensively used to provide solutions to the task assignment problem. These techniques are supported by useful software libraries such as pyschedule (github.com/timnon/pyschedule), PuLP (github.com/coin-or/pulp), Google OR-tools (developers.google.com/optimization), JuMP.jl (Dunning et al., 2017), Hungarian.jl (github.com/Gnimuc/Hungarian.jl) and CVXPY (www.cvxpy.org). These libraries support a variety of OR techniques including integer, linear, convex and dynamic programming.

Early techniques aimed at solving the task assignment problem by employing rigorous mathematical methods including integer, linear and dynamic programming (Cattrysse and Wassenhove, 1992). These techniques were able to deliver strict but CPU-intensive models. Furthermore, the provided solutions were not applicable to real world scenarios, since the global optimal solutions did not always reflect the reality.

Recent approaches adopt heuristic methods towards mitigating the inherent complexity of the abovementioned mathematical solutions. For instance, a hybrid approach is proposed in (Hong-Bing et al., 2002), aiming to solve the assignment problem by heavily exploiting neural networks. This approach relies on Hopfield and Tank's (Hopfield and Tank, 1985) observation that the underlying mechanism of a neural network intends to minimize a cost function by trying to find a local minimum. The proposed algorithm consists of two phases: (i) development of a neural network that finds a local minimum point, and (ii) an implementation of the dynamic tunneling technique, which assists in escaping from the local minimum. By iteratively running the above phases, the global minimum point (if any) corresponding to the best solution is discovered.

Bassett (2000) develops two approaches to optimize the utilization of employees' time and expertise. The first uses mathematical programming techniques, while the second one a heuristic approach to simulate the decision made by the project managers. The heuristic approach provides quick near-optimal solutions, contrary to the mathematical programming method which runs an exhaustive search to find the best solution.

On a broader resource allocation perspective, (Wang et al., 2018) exploits historical data to efficiently allocate radio resources in a telecommunication network. This approach searches

for optimal or near-optimal solutions of historical scenarios to adopt the most applicable to the current setting. For the selection of past similar scenarios, a classification approach is proposed which is based on the k-NN algorithm. In contrast to the classical greedy solutions for resource allocation, this approach avoids online calculations, hence augmenting the performance of this approach.

Finally, a genetic algorithm for resource allocation concerning construction projects has been developed in (Liu et al., 2005). The algorithm maps the allocation problem into the act of generating valid chromosomes. Each gene value in a generated chromosome corresponds to a construction activity. The outcome of the proposed algorithm allocates resources to the activities depending on the importance of each activity.

3 OUR APPROACH

It has been widely admitted that a company has more chances to succeed if its task assignment process considers the skills of each employee and the nature of the job to be accomplished in each task (Kelemenis and Askounis, 2010). Our approach aims to increase this very chance of success by properly assigning the available employees to pending tasks. For a given task X and an employee Y, we consider a *relevance metric* as the probability that the employee Y possesses the skills required by task X and, consequently, his/her capability of successfully completing the task on time (Hill et al., 2000). Our approach is divided in two phases (Figure 1):

- **Personnel Selection:** the estimation of how relevant/qualified each employee is to undertake and successfully accomplish each task. For the needs of this phase, diverse ML techniques are employed, which process and analyze textual data such as the 'title', the 'summary' and the 'description' of each task. This phase comprises three steps, namely 'Feature Extraction', 'Feature Generation' and 'Text Classification', which are described in detail below;
- **Human Resource Allocation:** the assignment of employees to tasks in a way that the total relevance metric is maximized. The linear assignment problem algorithm (Burkard et al., 2009) is utilized in this phase.

Our approach reveals hidden knowledge that exists in unstructured textual data. It is domain agnostic, while no additional information about the employees (e.g. resumes of employees) or the tasks (e.g. predefined keywords describing the required skills of each task) is needed to operate. Contrary to our approach, the already proposed solutions (i) process semi-structured or structured data (often obtained through questionnaires), (ii) rely on domain specific knowledge bases, and (iii) require information about each employee and task.

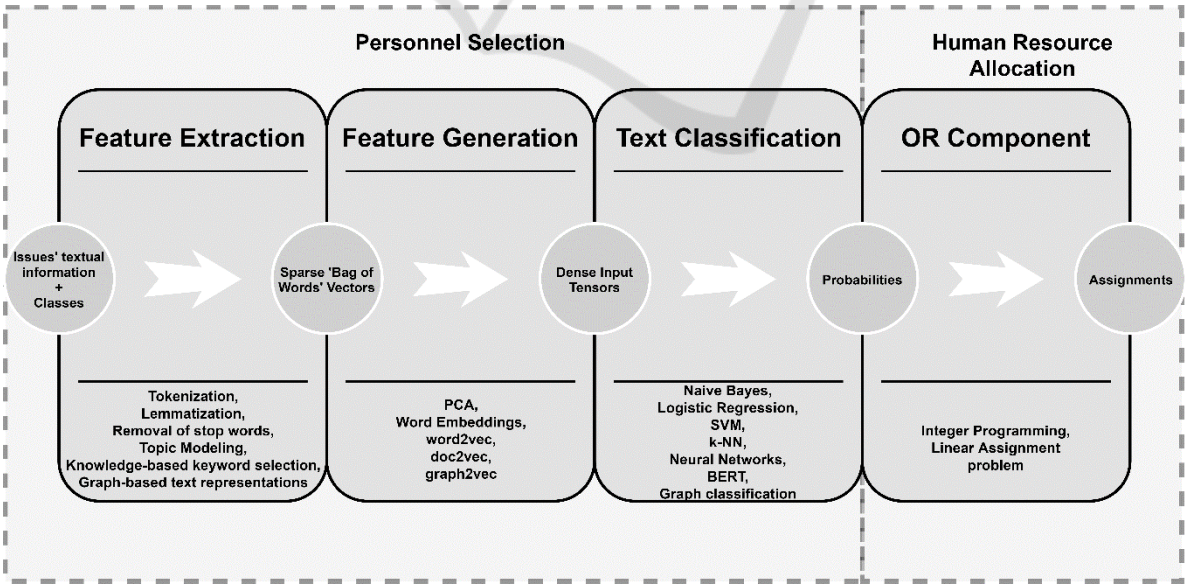

Figure 1: The proposed approach.

3.1 Feature Extraction

To extract features needed in the Text Classification step, a variety of Natural Language Processing (NLP) techniques, including feature generation and text normalization, is applied to the most important textual attributes of each task, namely 'summary' and 'description'. This step is accomplished through the following techniques that generate a 'Bag of Words' (BOW) vector:

- *Tokenization:* the process of generating tokens from unstructured text;
- *Lemmatization:* the process of grouping together the different inflected forms of a word;
- *Stemming:* the process of reducing derived words to their root form;
- *Feature/Tag generation:* the process of transforming an unstructured text into a vector that contains word occurrences;
- *Removal of stop words:* the process of removing commonly used words, which usually add noise to the ML models;
- *Removal of frequently occurred words:* the process of removing insignificant words from a text, taking into consideration the document frequency value of each word; in our example, we remove the words with document frequency greater than 60%.
- *Topic modeling:* the process of discovering abstract topics that occur in a task (e.g. 'bug fixing', 'development') (Rehurek and Sojka, 2010);
- *Knowledge-based keyword selection:* the process of identifying important keywords from the textual information of a task based on a lexicon/database of domain-specific predefined terms (Singh, 2017);
- *Graph-based text representation:* the act of representing a document as a graph to extract important features using classical graph algorithms such as node centrality and random walks (Nikolentzos et al., 2017).

3.2 Feature Generation

By exploiting the BOW vector generated from the Feature Extraction step and a common ML model (e.g. naive Bayes), we are able to accurately simulate the process of personnel selection made by the project managers of a company. However, in large datasets the produced vector representations suffer from the 'curse of dimensionality' phenomenon, which in turn contributes to (i) a-most-likely overfitting result, and (ii) a decrease in the model's accuracy. To mitigate the effects of the shortcomings of the BOW representations, the feature generation step employs dimensionality reduction and feature selection methods such as principal component analysis and word embeddings.

The outcome of the feature generation step is a 'dense input' tensor. This tensor integrates into its dense representation the most important features needed in order our model to make rational decisions.

3.3 Text Classification

As far as the calculation of relevance is concerned, we train a ML model that exploits the available textual information concerning each task (i.e. the 'summary' and 'description' attributes); contrary to existing solutions which utilize structured data, our proposition uses unstructured textual information to identify candidate employees. Hence, we reduce the relevance calculation problem to the common and well-studied *text classification* problem. Existing ML classifiers that are suitable for the text categorization problem include naive Bayes, logistic regression, support vector machines, k-nearest neighbors and neural networks, as well as more sophisticated solutions such as Google BERT (Devlin et al., 2018).

4 EXPERIMENTS

To test our approach, we extract and analyze data from the publicly accessible Jira instance of Apache Software Foundation (https://issues.apache.org/jira). This dataset concerns the development of 168 software projects including 'Hadoop', 'Spark' and 'Airflow'; it contains information related to 228,969 Jira issues. Each Jira issue in our dataset has the attributes 'summary', 'description', and 'assignee'.

The set of classes of our model corresponds to the names of the available employees ('assignee' attribute). Each task (i.e. Jira issue) is assigned to a class, i.e. the class with the highest probability (Mooney and Roy, 2000). Despite the fact that the classifiers assign only one class (i.e. employee) to a task, they also predict the probability that the task belongs to all other classes (i.e. how relevant is this task to each available employee). It is clear that in our approach the relevance metric for each task is equivalent to the probability calculated by the adopted ML models. Aiming to identify the statistically significant information, we keep only the Jira issues where their assignee has worked at least in 850 tasks (57,798 out of 228,969). As results, most of

Table 1: The task assignments per text classifier.

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
naive Bayes	E_{14}	E_7	E_{12}	E_9	E_4	E_{10}	E_3	E_6	E_2	E_1	E_0	E_5
logistic regression	E_{13}	E_{12}	E_5	E_9	E_{14}	E_{10}	E_3	E_2	E_{16}	E_1	E_0	E_{20}
SVM	E_8	E_7	E_{12}	E_9	E_{17}	E_{19}	E_{16}	E_4	E_2	E_{13}	E_{14}	E_5
k-NN	E_{13}	E_{18}	E_{12}	E_9	E_{14}	E_{10}	E_3	E_2	E_0	E_1	E_{17}	E_8
neural network	E_{13}	E_7	E_5	E_1	E_{14}	E_{16}	E_3	E_2	E_0	E_{17}	E_9	E_{20}

the remaining Jira issues have been assigned to 21 developers (noted below as employees E_1-E_{21}).

For the needs of the example described in this paper, we elaborate an application scenario using the textual information contained in 12 Jira issues of our dataset (noted below as tasks T_0-T_{11}). We test our approach by training five ML classifiers, namely 'naive Bayes', 'logistic regression', 'SVM', 'k-NN' and 'neural network'. Due to space limitations, we only provide here the outcome matrices (i.e. the task assignments) per text classifier (see Table 1).

The detailed input matrices (i.e. the relevance of each employee per issue) can be found at https://nkanak.github.io/icores-2020/examples. The code developed is available at https://github.com/nkanak/icores-2020. The dataset can be found at https://nkanak.github.io/icores-2020. Descriptive statistics of the dataset can be found at https://nkanak.github.io/icores-2020/descriptive-statistics. Finally, for the implementation of our experiments, we use the scikit-learn ML library (https://scikit-learn.org).

4.1 Naive Bayes

We adopt the multinomial naive Bayes classifier (scikit-learn, Python class: sklearn.naive_bayes.MultinomialNB) since it is suitable for classification with discrete features such as word counts.

4.2 Logistic Regression

The second classifier is the logistic regression (scikit-learn, Python class: sklearn.linear_model. LogisticRegression). Whilst the implementation of logistic regression is simple and easily interpretable, it frequently outperforms more complex ML classifiers.

4.3 SVM

The SVM (scikit-learn, Python class: sklearn. svm.SVC) classifier is effective when the number of features is more than the number of training examples.

4.4 k-NN

The k-NN (scikit-learn, Python class: sklearn. neighbors.KNeighborsClassifier) classifier is suitable for datasets where nonlinearity occurs. In our example, the hyperparameter k (i.e. the number of neighbors to use) is set to 16.

4.5 Neural Network

The final classifier is the neural network classifier (scikit-learn, Python class: sklearn.neural_network. MLPClassifier). Neural networks are widely used in classification problems dealing with unstructured data such as images and text. We trained a neural network that has 2 hidden layers with 1000 and 500 neurons respectively. We use the 'adam' solver as it accurately performs on large datasets.

4.6 Evaluation

4.6.1 Evaluation Metrics

Aiming to evaluate each text classifier, we use four common data mining metrics, namely 'accuracy',

weighted ‘precision’, weighted ‘recall’ and weighted ‘F1 score’ (see Table 2). A detailed explanation of the previous metrics can be found in (Aggarwal, 2015). Considering these metrics, we conclude that the (i) neural network (accuracy: 86%), (ii) logistic regression (accuracy: 86%) and (iii) naive Bayes (accuracy: 81%) classifiers sufficiently predict the employee for each task (it is noted that, when the ‘accuracy’ of a classifier is higher, the probability of how relevant an employee is to a task is more precise).

Table 2: Accuracy, weighted precision, weighted recall and weighted F1 score for each text classifier.

	Accuracy	Precision	Recall	F1
naive Bayes	0.81	0.84	0.81	0.80
logistic regression	**0.86**	0.85	**0.86**	**0.85**
SVM	0.53	0.59	0.53	0.39
k-NN	0.68	0.67	0.68	0.65
neural network	**0.86**	**0.86**	**0.86**	**0.85**

4.6.2 Explanations

To get a solid understanding of the underlying mechanism of our trained model, we explain the predictions of the model using the *Local Surrogate Models* (LIME) explanation method (Ribeiro et al., 2016; Guidotti and Monreale, 2019). In brief, this method tries to explain why single predictions of black-box ML classifiers were made by perturbing the dataset and building local interpretable models. In this case, the LIME text explainer randomly removes words/features from the text of each issue and calculates the importance of a specific word to the decision made by each text classifier.

The provided explanations help us check the reliability and validity of the trained ML models (Karacapilidis et al., 2017); they also confirm that the models select the right class for the right reason (i.e. meaningful words/features). For instance, Figure 2 explains why ‘elserj’ is selected to work on a specific Jira issue; this is due to the fact that features such as ‘protobuf’, ‘plan’ and ‘rpc’ play an important role in the decision made by the classifier. Additionally, features such as ‘website’, ‘format’, ‘docu’ (resulting from a processed version of the word ‘documentation’), increase the probability of ‘wesmckinn’ being the most suitable developer for the selected issue.

5 CONCLUSIONS

In this paper, we present a new approach for the task assignment problem. By applying NLP techniques to gain insights from unstructured textual data, we calculate how relevant an employee is to accomplish a specific task. Then, we maximize the total relevance by adopting a well-tried OR algorithm. We test and evaluate our approach using five classical text classifiers, namely naive Bayes, logistic regression, SVM, k-NN and neural network. The evaluation feedback is very promising, in that an accuracy score of 86% is achieved.

Future work directions concern: (i) the combination of various feature extraction, representation learning and text classification techniques; (ii) enrichment of the existing corpus of textual data using external knowledge (e.g. DBPedia

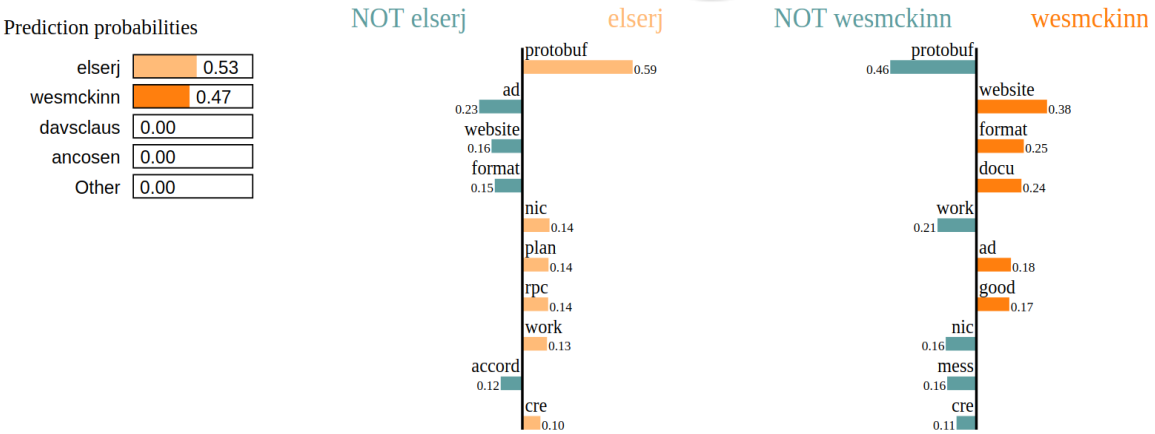

Figure 2: Explaining the selection of employees.

and Wikipedia knowledge), and (iii) integration of additional task features including priorities as well as budget and time constraints.

REFERENCES

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.

Alkhraisat, H. (2016). Issue Tracking System based on Ontology and Semantic Similarity Computation. *International Journal of Advanced Computer Science and Applications*, 7(11), 248-251.

Arias, M., Saavedra, R., Marques, M. R., Munoz-Gama, J., & Sepúlveda, M. (2018). Human resource allocation in business process management and process mining: A systematic mapping study. *Management Decision*, 56(2), 376-405.

Azzini, A., Galimberti, A., Marrara, S., & Ratti, E. (2018). A classifier to identify soft skills in a researcher textual description. *In International Conference on the Applications of Evolutionary Computation* (pp. 538-546). Springer, Cham.

Bassett, M. (2000). Assigning projects to optimize the utilization of employees' time and expertise. *Computers & Chemical Engineering*, 24(2-7), 1013-1021.

Burkard, R. E., Dell'Amico, M., & Martello, S. (2009). *Assignment problems*. Philadelphia.

Cattrysse, D. G., & Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European journal of operational research*, 60(3), 260-272.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dunning, I., Huchette, J. and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2), pp. 295-320

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5), 93.

Heyn, V., & Paschke, A. (2013). Semantic Jira-Semantic Expert Finder in the Bug Tracking Tool Jira. *arXiv preprint arXiv:1312.5150*.

Hill, J., Thomas, L. C., & Allen, D. E. (2000). Experts' estimates of task durations in software development projects. *International journal of project management*, 18(1), 13-21.

Hong-Bing, X., Hou-Jun, W., & Chun-Guang, L. (2002). A hybrid algorithm for the assignment problem. *In Proceedings of the International Conference on Machine Learning and Cybernetics* (Vol. 2, pp. 881-884). ACM.

Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological cybernetics*, 52(3), 141-152.

Kanakaris, N., Karacapilidis, N., & Lazanas, A. (2019). On the Advancement of Project Management through a Flexible Integration of Machine Learning and Operations Research Tools. *In Proceedings of the 8th International Conference on Operations Research and Enterprise Systems*. (pp. 362-369). SciTePress Publications.

Kanakaris, N., Karacapilidis, N., Kournetas, G., & Lazanas, A. (2020). Combining Machine Learning and Operations Research Methods to Advance the Project Management Practice. In *International Conference on Operations Research and Enterprise Systems* (pp. 135-155). Springer, Cham.

Karacapilidis, N., Malefaki, S., & Charissiadis, A. (2017). A novel framework for augmenting the quality of explanations in recommender systems. *Intelligent Decision Technologies*, *11*(2), (pp. 187-197).

Kelemenis, A., & Askounis, D. (2010). A new TOPSIS-based multi-criteria approach to personnel selection. *Expert systems with applications*, 37(7), 4999-5008.

Liu, Y., Zhao, S. L., Du, X. K., & Li, S. Q. (2005). Optimization of resource allocation in construction using genetic algorithms. *In 2005 International Conference on Machine Learning and Cybernetics* (Vol. 6, pp. 3428-3432). IEEE.

McDonald, D. W., & Ackerman, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. *In Proceedings of the 2000 ACM conference on Computer supported cooperative work* (pp. 231-240). ACM.

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. *In Proceedings of the fifth ACM conference on Digital libraries* (pp. 195-204). ACM.

Nikolentzos, G., Meladianos, P., Rousseau, F., Stavrakas, Y., & Vazirgiannis, M. (2017). Shortest-path graph kernels for document similarity. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1890-1900).

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*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. *In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144). ACM.

Singh, V. (2017). Replace or Retrieve Keywords In Documents at Scale. *arXiv preprint arXiv:1711.00046*.

Sullivan, C. A., Zimmer, M. J., & Richards, R. F. (1988). *Employment Discrimination* (pp. 266-n). Boston: Little, Brown.

Wang, J. B., Wang, J., Wu, Y., Wang, J. Y., Zhu, H., Lin, M., & Wang, J. (2018). A machine learning framework for resource allocation assisted by cloud computing. *IEEE Network*, 32(2), 144-151.

Wowczko, I. (2015). Skills and vacancy analysis with data mining techniques. *In Informatics* (Vol. 2, No. 4, pp. 31-49). Multidisciplinary Digital Publishing Institute.