Computer Science and Information Systems 21(1):75–94 https://doi.org/10.2298/CSIS220922065A

Machine Learning and Text Mining based Real-Time Semi-Autonomous Staff Assignment System

Halil Arslan[1], Yunus Emre Işık[2], Yasin Görmez[2], and Mustafa Temiz[2]

[1] Department of Comuter Engineering,Sivas Cumhuriyet University
58140 Sivas, Türkiye
harslan@cumhuriyet.edu.tr
[2] Department of Management Information Systems, Sivas Cumhuriyet University
58140 Sivas, Türkiye
{yeisik,yasingormez,temizmustafa}@cumhuriyet.edu.tr

Abstract. The growing demand for information systems has significantly increased the workload of consulting and software development firms, requiring them to manage multiple projects simultaneously. Usually, these firms rely on a shared pool of staff to carry out multiple projects that require different skills and expertise. However, since the number of employees is limited, the assignment of staff to projects should be carefully decided to increase the efficiency in job-sharing. Therefore, assigning tasks to the most appropriate personnel is one of the challenges of multi-project management. Assigning a staff to the project by team leaders or researchers is a very demanding process. For this reason, researchers are working on automatic assignment, but most of these studies are done using historical data. It is of great importance for companies that personnel assignment systems work with real-time data. However, a model designed with historical data has the risk of getting unsuccessful results in real-time data. In this study, unlike the literature, a machine learning-based decision support system that works with real-time data is proposed. The proposed system analyses the description of newly requested tasks using text-mining and machine-learning approaches and then, predicts the optimal available staff that meets the needs of the project task. Moreover, personnel qualifications are iteratively updated after each completed task, ensuring up-to-date information on staff capabilities. In addition, because our system was developed as a microservice architecture, it can be easily integrated into companies' existing enterprise resource planning (ERP) or portal systems. In a real-world implementation at Detaysoft, the system demonstrated high assignment accuracy, achieving up to 80% accuracy in matching tasks with appropriate personnel.

Keywords: multi-project management, task assignment, text mining, staff assignment system

1. Introduction

The continuous development of technology and information systems is rapidly changing the business mentality in the global world. Companies have started to use information systems, customized applications, and software in all departments such as production, logistics, marketing and human resources in order to reduce costs, gain a competitive advantage and make the organization more efficient. The need to transform and update existing business processes in companies using information-based systems has emerged with

the introduction of public obligations. However, efforts to meet Information Technology (IT) requirements in large organizations with internal resources may not be sustainable, as complex projects require different levels of expertise. Instead, outsourcing projects to consulting firms that have expert staff offers benefits in terms of focusing on the main area of work, increasing quality, and reducing costs [29].

On the other hand, the workload of consulting firms has increased as the number of clients with diverse businesses who are awaiting the services of their consultants for the new project increased. In addition, needs analysis, action plan, resource allocation, and testing procedures should be scheduled for the requested project/task support [13]. Timely completion of project tasks requiring different skills is also possible only with multi-project management (MPM).

MPM is an approach that involves planning, executing, monitoring, and completing multiple projects simultaneously using the same set of human resources. The objective of this management approach is to achieve optimal organizational performance by effectively balancing and coordinating projects that require specialized expertise while dealing with limited resources[11]. In cases where the most important resource is educated and qualified personnel, the optimal output can only be achieved by assigning the right personnel to the right project. This challenge is called a task assignment problem in the MPM environment. Task assignment is about finding the most suitable personnel who have the required skills to perform the tasks in a new or existing project. When determining personnel, it is expected that the employee's qualifications match the requirements of the task. Otherwise, the non-appropriate assignment of personnel might lead to delays in projects. Since projects in IT consulting are generally software or development tasks, each of them must be carried out by a qualified person according to the needs of the task. Even if there are many employees in a company, not every employee may have the same skills and experience. Therefore, there is a need for solutions that efficiently facilitate task assignment is crucial for enhancing productivity and maximizing the utilization of qualified human resources.

The existing studies in the literature treat the assignment of tasks and employees in MPM as scheduling and optimization of resources depending on various objective functions [15], [27],[24],[16]. Besides there are some studies that tried to solve real problems of companies. Lie et al. proposed the Critical Chain Project Management approach to efficiently manage projects in research centers. In this approach, resources are divided into small parts, and procurement of required equipment is included in the project management process, which increases efficiency [20]. Chen et al. proposed an integrated model for scheduling multiple projects and assigning staff with multiple skills for IT products. They defined four objectives to be considered simultaneously: Skill efficiency, product development, time and cost. The non-dominated Sorting Genetic Algorithm II is used to solve the optimization problems [8].

One of the most important challenges of MPM is the problem of efficient task assignment [17]. Task assignment is about finding the most suitable personnel who have the required skills to perform the tasks in a new or existing project. When determining personnel, it is expected that the employee's qualifications match the requirements of the task. In a study, Cai and Li proposed a genetic algorithm-based model for assigning employees with multiple skills to the right tasks [5]. Cheng and Chu proposed a model that considers employee qualifications and optimizes the task assignment problem using fuzzy set the-

ory and genetic algorithm [9]. Almost all of these studies aim to demonstrate the success of optimization models in theoretical terms by testing them on fixed datasets. Since these approaches rely on constraints, they must be repeated whenever a new task is requested. Almost all projects run by companies have a deadline. Projects that cannot be completed by the deadline cause companies to incur huge losses. The fastest way to assign appropriate employee, which is one of the most important factors for projects, also significantly affects the timely completion of the project. In this context, task assignments should be completed immediately when a new task is requested. The re-optimization of the models can take a lot of time depending on the size of the data used. Considering that decision makers need to make the assignment process quickly, the models in the literature are not suitable for a real-time system. Contrary to this situation, there is no need for re-training in the system that was proposed in this study.

A machine learning solution incorporates a text-mining approach, on the other hand, can be used to recommend efficiently a staff once it has been properly trained. After all, given that project tasks are identified and described through textual information, text mining emerges as one of the most effective ways for analyzing such textual data and deriving valuable insights from it. Text mining, also known as text analysis or text data mining, is a field that combines multiple disciplines to extract valuable information and knowledge from unstructured text data [30]. Utilizing pre-processing techniques and algorithms, enables automatic processing and interpretation of given text data, leading to several advantages over manual content analysis, such as less required time and human work needed. [14]. Mo et.al. proposed a model to improve the productivity of staff assignments using text mining-based machine learning techniques. More than 82,000 collected maintenance records from different universities were vectorized using Bag-of-Words (BOW). Logistic Regression, Support Vector Machines, and Naive Bayes methods are used to train the datasets. The models are then used to determine the personnel for specific tasks. As a result, 77% of the tasks are correctly assigned to the correct personnel [23]. A similar machine learning-based study was conducted to route service requests to the help desk system to the correct staff. In the study, using algorithms such as SVM, Decision Tree, and Naive Bayes, service request tickets are classified and routed to the right staff with 81% Accuracy [3].

Appropriate personnel assignment studies are directly related to companies working on more than one project at the same time. It is possible that decision support systems designed using datasets created from historical data may give worse results in real-time data. In this context, the fact that a study on personnel assignment also works with real-time data of companies is one of the important factors that increase the value of the study. Especially for companies operating in the field of consultancy, assigning personnel using real-time data has crucial importance. Companies that manage multiple projects are generally medium or large-sized companies, and these companies mostly run their projects through ERP systems. In this study, real-time data obtained from the ERP system of a software consultancy firm operating in Turkey were used. The performance obtained using real-time data with the proposed model in this study has similar results with the analyzes made using historical data in the literature. In fact, better results have been obtained from many models that analyze using historical data in the literature. Considering this structure, our study differs positively from the studies in the literature.

2. Materials and Methods

2.1. Grounded Theory

In qualitative research methods, the establishment of theories about data can lead to the collection of specific data that validate that theory. If there are not enough theories in the literature on a topic, deductive reasoning may not be the best way to find out true theories about data. In order to draw meaningful conclusions about the collected data on any topic, theories should be made about the existing data. In statistical analysis, this approach is called grounded theory. Grounded Theory (GT) is a qualitative research method based on the systematic construction of hypotheses and theories through the collection and analysis of qualitative data [7]. It provides summarized ideas and concepts from the collected data. GT follows 4 iterative rules: Finding representative concepts by reviewing the data, recoding concepts using keywords, hierarchically grouping the codes into concepts, and categorizing them by similarity. At the end of the iterative analysis, the categories and the connections between them are used as theory.

2.2. TF/TFIDF

The most commonly used text vectorization methods in the literature are Term Frequency (TF) and Term Frequency-Inverse Document Frequency Ratio (TF-IDF). TF simply indicates the ratio of the occurrence of each word to the total number of words and is calculated by dividing the frequency of occurrence of a word by the number of words in the document [22]. Thus, if the ratio of a word is higher, it is considered as an important and prominent word with respect to the topic of the document. On the other hand, TF-IDF is calculated by multiplying the inverse document frequency, which indicates the importance coefficient of a word in a given document, by the term frequency. Thus, TF-IDF considers not only the word frequency but also the importance coefficient of a word in a document to indicate the coefficient. Since these approaches are most commonly used, we tested them in our dataset and compared them with word embedding methods [31].

2.3. Word2Vec

Word2Vec, which is a word embedding approach, is a technique that maps words to a vector of numbers. It uses a two-layer neural network model to understand the semantic relationship behind words in a given context, and hypothesizes that words with close meaning also have close vectorial distance [21]. This vectorization method has 2 different learning algorithms, CBOW (Continuous Bag of Words) and Skip-Gram, as shown in figure 1.

CBOW receives a couple of words $W_n = W_t - 2, W_t - 1, W_t + 1, W_t + 2$ as input, where n denotes the window size of words and W_t denotes the target words. The main principle of CBOW is to predict a particular word by analyzing the neighboring words. On the other hand, Skip-Gram attempts to predict the surrounding words from the target words, as a reversal of CBOW. Skip-Gram takes advantage of vectorization when a new word appears in context. The projection layer in both models is an N-dimensional vector projected by an encoded input vector. This layer stores a single set of common weights and therefore projects all words to the same position.

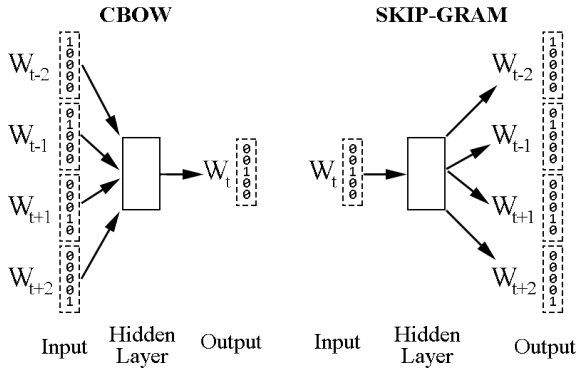

Fig. 1. Learning Models of Word2Vec Representation. The 't' represents the current word. In CBOW, a window of context is employed to predict the target word by considering both preceding and succeeding words. In contrast, Skip-Gram aims to predict the previous and next words using the middle word as the context

2.4. Doc2Vec

The Word2Vec method attempts to discover similarity between words by using co-occurrence frequencies and vector distances, but the Doc2Vec approach represents the entire document or paragraph in a vector space rather than just words, regardless of text length [19]. Therefore, it is considered as an extension of the Word2Vec approach. In this method, additional identifier information is added to the vector so that paragraph-to-paragraph relationships can be learned and models can understand the similarities between paragraphs behind words.

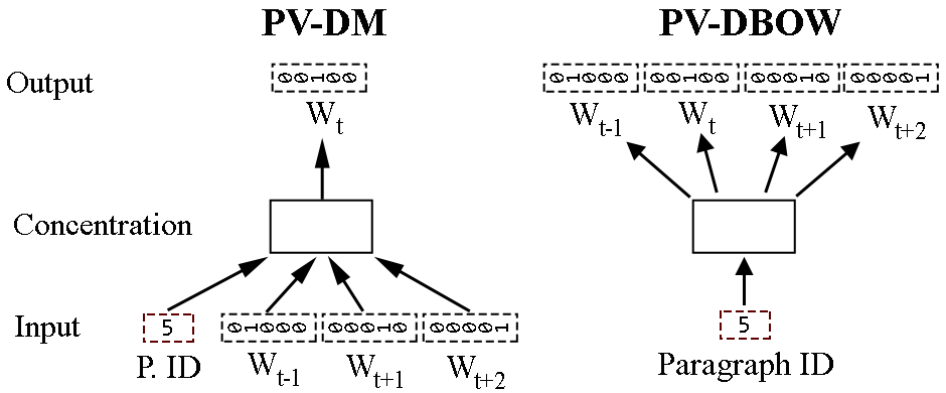

Fig. 2. Learning Approaches of Doc2Vec Representation. Similar to Word2Vec learning but includes an extra information paragraph id

The Doc2Vec representation method includes PV-DM (Paragraph Vector-Distributed Memory) and PV-DBOW (Paragraph Vector-Distributed Bag of Words) like the CBOW and Skip-Gram learning algorithms. PV-DM enables paragraph-specific word learning by using the unique identity value of the paragraph along with the word vectors. On the other hand, PV-DBOW does not take a word vector but the unique identification number of the paragraph and tries to predict the target words (see Figure 2).

2.5. FastText

FastText is a text mining library developed by Facebook group for text classification and word representation. The algorithm behind FastText is slightly different from other word representation methods such as Word2Vec and Glove [4]. The FastText algorithm treats n-grams at the character level of a word as the smallest unit. Using special delimiter symbols at the beginning and end, words are divided into subwords of length n. For example, the FastText representation of the word "detay" is ¡de,det,eta,tay,ay¿ if the n-gram is equal to 3. After the composition of the n-gram subwords, the embedding vector of the word is calculated as the average of these n-gram vectors. The remaining word learning processes are similar to those of Word2Vec, where the neighboring words of the context are learned using the skip-gram approach. FastText also has a text classification module that uses a neural network to predict the class labels of input documents. The mean values of the word embedding vectors form the document vector for the neural network model. After the model is trained, documents or any text sentences can be classified using the model.

2.6. ITU NLP Tool

One of the main challenges in natural language processing studies is that each language has its own unique linguistic structure. For example, in agglutinative languages, word stems often change and become unrecognizable when new words are derived, while in polysynthetic languages, words are conjugated by adding prefixes/suffixes that do not cause stem changes [6]. Therefore, natural language processing algorithms should be developed in a language-specific manner to improve performance. ITU NLP Tool (Istanbul Technical University Natural Language Processing Tool) is a web-based service specified for Turkish languages and includes various linguistic analysis modules such as Tokenizer, Normalizer, Morphological Analyzer, etc. [12]. The application analyzes and parses the given text with different modules and uncovers linguistic word components in the text, such as word stems, conjunctions, adverbs and adjectives. To clean and parse the words in the dataset, we used the tool ITU NLP in our experiments.

2.7. Cosine Similarity

To perform a comparison of vectors, we need a measure that can calculate the similarity of given vectors. Cosine similarity is one of the well-known methods to determine the similarity between vectors that are not zero in an inner product space. Mathematically, it measures the cosine of the angle between two vectors projected into a multidimensional space, and the result is clearly limited to [0,1].

$$cos(\theta) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

The cosine similarity of two given vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is calculated by formula (1), where $\boldsymbol{a} \cdot \boldsymbol{b}$ is the dot product of the vectors and $||\boldsymbol{a}||$, $||\boldsymbol{b}||$ are the length of the two vectors. If the result of $cos\theta$ is equal to 1, it means that the given vectors are exactly equal [18].

3. Proposed Approach

Our system is designed to predict the most suitable staff when a task request is received through the portal system. Assigning a task that requires specific qualifications to a staff member who lacks adequate experience in those qualifications may result in delays, non-completion, and financial losses. To address this issue, it is crucial to assign the most qualified personnel to tasks. However, this approach may lead to imbalances in job-sharing among staff and conflicts arising from simultaneous tasks across different projects. Therefore, the system should offer multiple staff members, sorted based on their ability to meet the task requirements. For instance, if a web development task requires design alterations, a staff member specializing in front-end development could be sufficient. However, for the development of a complete website, it is more efficient in terms of cost to assign a staff member with experience in both back-end and front-end development. By considering the specific requirements of each task, our system aims to optimize staff allocation and ensure task completion in a timely and cost-effective manner. In order to ensure that our system compares needs of task and staff-wise qualification.

The system comprises two primary steps to fulfill its functionality. Firstly, the task text is processed to identify the required skills, such as development, help desk, RD (Research and Development). The task management module serves as a repository, storing comprehensive information including task descriptions, subjects, and project names. This module integrates with the company's ERP systems and forwards the received tasks to the text mining module through the "Task Repository," which acts as the database for the management module.

Text Mining module performs to text data a series of operations such as the removal of tags and punctuation. These operations ensure data cleanliness of the text data. We also tried different types of pre-processing using the text mining methods such as stopwords removal, word normalization and stemming as described in the dataset section. The other key point has effect on performance is vectorization of the text data. Vectorization is converting process of a data type to vectors that enables to analyses of textual data with machine learning. Within our study, we experimented with 4 different vectorization methods, namely TF, TFIDF, Word2Vec and Doc2Vec to compare the effect of different vectorization methods and select the best method for the live system. The Python library Scikit-Learn was used to implement the TF and TFIDF vectorization approaches [25]. These methods do not require a learning algorithm since they vectorize the words based on the text statistics.

The Word2Vec and Doc2Vec methods are essentially a neural network model with an input, a hidden, and an output layer that attempts to uncover the semantic relationship between words. It is necessary to train these models to learn the semantic similarities specific to the data. However, it is crucial to avoid utilizing test data during the optimization

process to prevent overfitting and ensure fair predictive performance of the model. Otherwise, it is concluded that the results to be obtained are not reliable. Hence, the training dataset is re-splitted into 80% validation training and 20% validation testing. Figure 3 summarizes all the steps of our proposed system.

The second main step of proposed system involves identifying the optimal staff member for a classified task. The Staff Qualification module, which contains personnel information and the up-to-date qualifications, is used when deciding by system on the assignment of new tasks to an employee who is the most suitable. The system utilizes 11 valued vectors, corresponding to class distribution, to represent the qualifications of individual staff members. After receiving the predicted task text vector from the classification modules and the personnel qualification vectors from the Staff Qualification module, the Decision Support System is activated to compare these vectors. In our approach, we employed cosine similarity for this comparison. Cosine similarity allows for a value-wise comparison between vectors, where each value corresponds to a specific label. For example, value corresponding to ”Abap” label is compared to ”Abap” value of prediction. However, total similarity is calculated by summing the element-wise similarities. For instance, if a task involves full web development along with some data mining analysis, our machine learning model possibly predict a vectors with higher values for ”ReDe” and ”Hybr” classes. Because data mining and web developments are concerned with ”ReDe” and ”Hybr” classes, respectively. Therefore decision support module should recommend some staffs who have experience for both of classes. Cosine similarity will calculate higher score to a staff member whose qualification vector has higher values for both ”ReDe” and ”Hybr” compared to a staff member who only excels in ”ReDe” but has lower values for ”Hybr”. This approach ensures that our model predominantly recommends the most suitable staff members, thereby optimizing resource allocation and reducing costs. Once a task is assigned to the staff by the system, upon its completion, the personnel vector is updated based on the predicted task text vector obtained from the machine learning model. As personnel specializing in one or more areas successfully accomplish tasks, their qualifications are also updated, enabling the company to monitor and track the continuous improvement of its personnel.

4. Experiment Results

Our live system runs with initialization of modules, prediction of related fields and estimation of the most suitable staff to the user for a newly received task from projects of the consulting company. To obtain a reliable and stable estimation, the system needs to predict the field for a given task text as accurately as possible, and this is only possible if the machine learning models have high predictive performance. Therefore, in an experimental step, different approaches for text vectorization, machine learning, and preprocessing were compared.

The first stage of experimental work includes data organization and preprocessing steps. The process of generating datasets involves four main steps. The first step is to collect and tag project/task requests. The request tickets contain real-time project/task requests from Detaysoft [1] which is one of the largest SAP consulting companies in Turkey. The company uses its portal system to manage tasks, resources and business. Requests are created and submitted by clients, managers or team leaders through the portal, along with

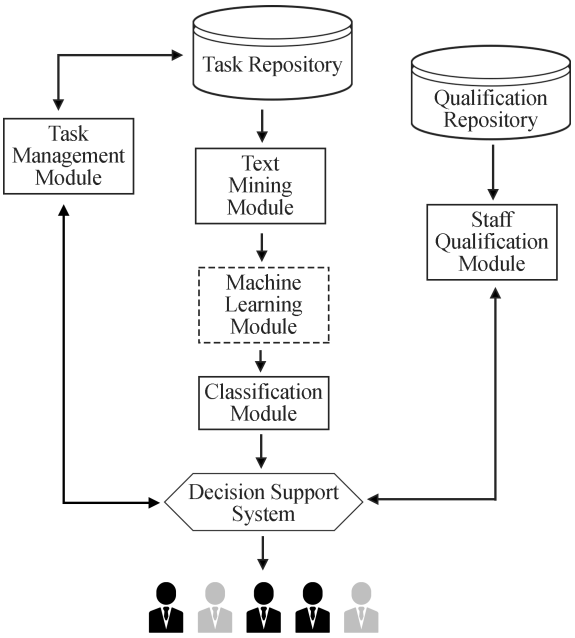

Fig. 3. Architecture of end-to-end Decision Support System

the selection of employees to be assigned the tasks. The collected request tickets are then downloaded, and irrelevant parts such as XML or HTML tags are removed. The result is a dataset with 2103 texts explaining tasks.

The second step is to label the relevant data. Since the company does not have an existing labeling pool, a qualitative research method called grounded theory is used to establish the most accurate labeling pool. Project managers and team leaders processed the raw data according to predefined rules and defined 11 class labels to label the samples. These classes represent SAP installation, update and configuration processes ”Bsis,” software development language ”Abap,” human resources processes ”HrGn,” e-commerce platform operations ”Hybr,” and data processing and reporting processes ”BwBo.” Logistics processes are represented by three subclasses: Sales-Distribution ”LoSd”,” Quality Management ”LoQm” and Materials Management ”LoMm” Financial transactions are represented by the labels ”FiCo” for cost accounting and ”FiRe” for financial property management. The label ”ReDe” is assigned to jobs that fall outside these 10 defined classes and usually require special expertise.

The third stage is the assignment of employee qualifications based on 11 predefined class labels by Grounded Theory. Machine learning models were also used to perform the assignment process systematically and based on the tasks performed by each employee. In this model, the employee qualifications dataset labeled by the team leaders was used. With this model, it is aimed to predict in which areas a staff whose competencies are given in writing is prone to develop projects. Since the team leaders will make improvements in the data labeling part, this model used in data labeling has only been used to speed up the data labeling process. In this model, Word2Vec and LR were used and

79% accuracy was achieved. For every completed task, a trained machine learning model predicts a qualification vector comprising the 11 class distributions, which is then set as the qualification distribution of the respective staff member. This approach enables the measurement of staff experience and qualifications in different fields. However, since the prediction performance of machine learning models is not 100%, the skill vectors were set as changeable by the team leaders or the employees themselves against wrong values.

The last step involves the generation of different datasets through various preprocessing techniques. In the text mining literature, it is well known that the use of unnecessary words, word normalization, and stemming have positive or negative effects on model success [32]. To show the impact of text preprocessing on machine learning prediction and overall system success, the raw data was preprocessed by removing stop words, normalizing words, and stemming words in ITU NLP tool, and then stored separately for the experiment (see Figure 4).

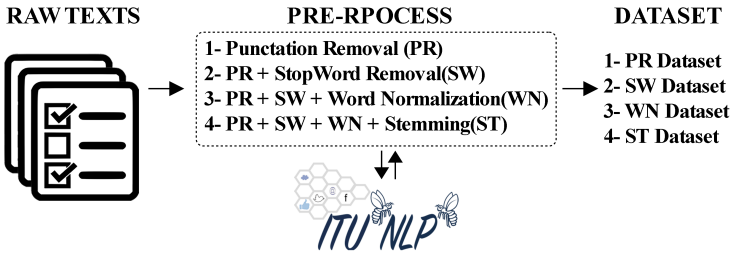

Fig. 4. Creation Steps of Different Dataset based on pre-processing

After pre-processing steps, dataset was divided into two parts to train and to test proposed model. The numerical distribution of the data used for the training and testing process in the models is shown in Figure 5. The data size of the training/testing sets was balanced to avoid class-specific overfitting and to ensure the model learned for all classes. Performance metrics are calculated by comparing the actual label of the task text in the dataset with the predicted class label. For instance, if a test task belongs to the ”ReDe” class and our model predicts it as ”ReDe”, it is considered a truly predicted sample. However, as previously mentioned, our model predicts an 11-valued vector prediction instead of a single class. To provide a fair assessment of the vector-based predictions, we also calculated the AUC score, which takes into account the overall performance of the vector values. This allows for a more comprehensive evaluation and comparison of the model’s predictions.

In the next phase, features were extracted using TF, TFIDF, Word2Vec and Doc2Vec. As mentioned before, TF and TFIDF methods do not need training, but Word2Vec and Doc2Vec methods need a training process as they are artificial neural network models.In order to train Word2Vec and Doc2Vec models and find optimal hyperparameters the GENSIM[28] and Optuna [2] libraries was used, respectively. Detailed information about the parameter spaces for these two models can be found in Table 1. As mentioned earlier, the main purpose of distributed word representation methods is to determine the vectors of words based on their semantic similarities. However, our goal in this study is to classify

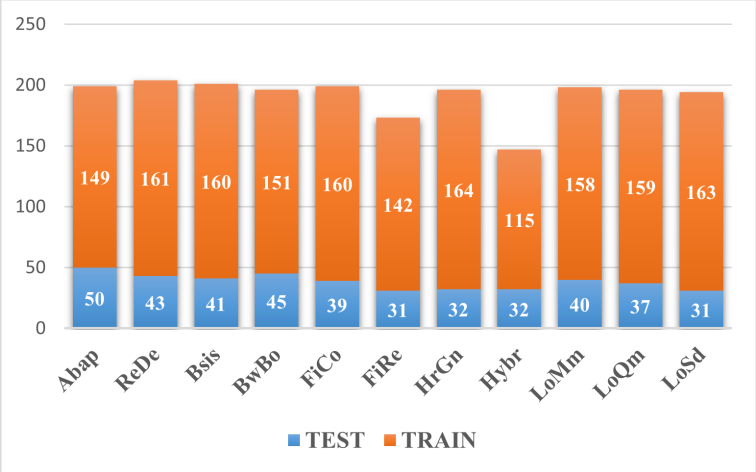

Fig. 5. Numerical distribution of samples in training and testing phase

the text of a task sample that contains more than one word, and even sentences. Therefore, it is necessary to calculate the vector of the sample from word vectors. To overcome this challenge, it is chosen to use word vectors learned using word embedding methods, and the vector of each task sample was formed by averaging the word vectors it contains. A random set of parameters from the given space is automatically set in the model for each iteration to perform a training and testing process in the optuna library. Therefore, the validation data for the hyper-parameter optimization of Word2Vec and Doc2Vec are reformed after each iteration depending on the parameter group.

Table 1. Parameter Space of Word2Vec and Doc2Vec representation methods for hyper-parameter optimization

Parameter	Parameter Space
Vector Size, Dimensionality of the word vectors	(5 - 5000)
Word Window, distance between current predicted words	(2 - 10)
Alpha, The initial learning rate	(0.00001, 10)
Epoch, number of iteration over dataset	[50,100,250,500,750,1000]
Training algorithm: 1 for skip-gram; otherwise CBOW.	[0,1]
Minimum Count, Min. Number of occurence of words.	(1 – 10)

Following the vectorization procedures applied to the raw text samples, the datasets were prepared for utilization in machine learning models. Table 2 provides an overview of the vector lengths for each sample across various vectorization approaches. Notably, the Word2Vec and Doc2Vec-based vectors exhibit slightly shorter lengths due to the nature of these methods, which involve embedding word representations.

Table 2. Vector lenghts of samples after each pre-process methods

Pre-Process Method	TF	TFIDF	W2V	D2V
Word Normalized Data	12969	12969	4178	2227
Word Stemmed Data	5554	5554	1546	4309
Stop Words Removed Data	13708	13708	4995	812
Punctation Removed Data	13919	13919	2899	4891

In addition to pre-processing methods, choosing the most successful classification algorithm for the live application system is also crucial for assigning tasks to the optimal personnel. Hence, our study involved a comparison of five different machine learning algorithms: logistic regression (LR), support vector machines (SVM), random forest (RF), k-nearest neighbor (KNN), XGBoost (XGB), and FASTText. Each dataset generated using various approaches was trained and tested with these algorithms except FASTText. Because, FastText classification module automatically converts the texts to vectors using its own Word2Vec method. Therefore, no other vectorized datasets were used for FastText, only preprocessed data. Subsequently, the best-performing model was selected and integrated into the end-to-end system. We also optimized hyperparameters due to reason of algorithms has a significant impact on predictive performance. To make our system more robust, the hyperparameters of the different algorithms were optimized separately using the Optuna library. The space and list of hyperparameters of each algorithm that were optimized are shown in Table 3.

Table 3. Parameter Space of machine learning algorithms for hyper-parameter optimization

Classification Algorithm	Parameter	Parameter Space
LR	Regularization (C)	$2^{-12} - 2^{12}$
	Solver Algorithm	Linear, BFGS
SVM	Regularization (C)	$2^{-12} - 2^{12}$
	Kernel	Linear, RBF, Polynominal
	Gamma	(0.000001 - 10)
RF	Number of Trees	(2 - 1000)
	Criterion	Gini, Entropy
KNN	Number of Neighbors	(2 – Number Of Samples)
	Algorithm	KD-Tree, Ball-Tree, Brute
XGB	Number of Trees	(2 - 1000)
	Learning Rate	(0.0001 - 10)
	Alpha	(0 - 32)
	Gamma	(0 - 32)
FASTTEXT	Vector Size	(3 - 5000)
	Window Size	(1-15)
	Length of Word n-gram	(1-5)
	Training Algorithm	[CBOW, Skip Gram]
	Epoch	[50,100,250,500,750,1000]
	Learning Rate	(0.00001 - 1)

Once the optimal parameters were found, the models were trained using only the training data and their performance was calculated using the test data. The workflow during the classification steps, starting with the partitioning of the data set and ending with the model tests, is illustrated in Figure 6. For each vectorization and classification process, the data was split in the same way and similar processes were performed to allow a fair comparison between the algorithms. As a result of the classification step, the best preprocessing, vectorization and classification methods in terms of working time, accuracy and ease of implementation were decided and included in the text processing and classification modules of the live system, respectively.

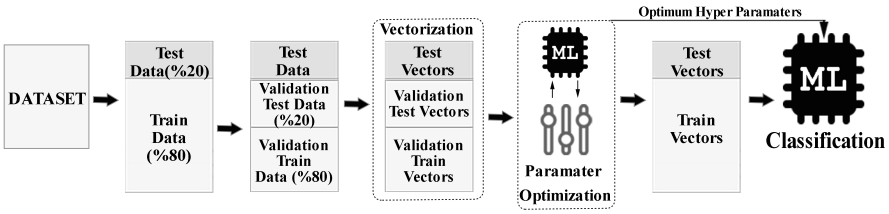

Fig. 6. Flow of validation and test process in machine learning experiments

The output of the classification module provides a prediction of what qualifications or skills are required for a new task. Since 11 different classes are used in labeling the data, the module outputs a vector containing 11 values for the class distributions. For example, the probability distribution vectors predicted by the classification module for 4 different test samples are shown in Table 4. The skill requirement of the first sample is predicted to be more in the ”Lomm” class at 62.9%, and also has an ignorable relationship with the ”Abap” field. The second sample, on the other hand, seems most likely to be related to the ”Abap” department and therefore assigning ”Abap” skilled employees to this task would be the best choice. However, it is clear that the third task requires a multi-skilled employee who should have knowledge for both ”FiCo” and ”LoSd” as deduced from the task text. Therefore, the use of an employee who has knowledge in both areas offers the greatest possible benefit in terms of multi-project management.

Table 4. Output samples of machine learning model that contains 11 class label distribution vector

ID	Class Labels (Departments)										
	Abap	ReDe	Bsi	BwBo	FiCo	FiRe	HrGn	Hybr	LoMm	LoQm	LoSd
1	0.17	0.01	0.02	0.05	0.03	0.00	0.02	0.06	0.63	0.01	0.01
2	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.01
3	0.08	0.05	0.01	0.02	0.50	0.02	0.02	0.04	0.01	0.03	0.22
4	0.07	0.01	0.02	0.08	0.01	0.04	0.03	0.35	0.35	0.02	0.02
...	...	...	...	...	...	...	...	...	...	...	...

Table 5 shows the results of the machine learning models for ”raw data,” i.e., no pre-processing was done except for tag removal. It can be seen that the TF-IDF vectorization method achieves the best results for all algorithms when analyzing the ”raw data”. The Word2Vec method, on the other hand, seems to be the most unsuccessful representation method for ”raw data”. Since the word embedding algorithms focus on the neighborhood relationship of words in the sentence, the models may have learned the relationship between stop words rather than between keywords, and therefore may not have predicted the correct vector representation that contains characteristic features related to classification into the correct class.

Table 5. Accuracy and AUC scores of machine learning algorithms on Raw Dataset which no any pre-processed applied

Method	Accuracy				AUC			
	TF	TFIDF	W2V	D2V	TF	TFIDF	W2V	D2V
LR	0.7625	0.7886	0.5843	0.7648	0.9594	0.9697	0.9058	0.9611
SVM	0.7458	0.7981	0.5487	0.715	0.9578	0.9721	0.8941	0.9563
KNN	0.1639	0.6746	0.4774	0.6437	0.8072	0.9452	0.87	0.9226
RF	0.7268	0.715	0.5938	0.6603	0.9549	0.9506	0.9106	0.9366
XGB	0.6817	0.6461	0.5677	0.5582	0.9339	0.9206	0.9063	0.8824

The Word2Vec models achieved the best predictive performance among all methods and models, with an accuracy of 80.52% in the dataset containing no stopwords (see Table 6). This result also proves that stop words have the opposite effect of the word embedding methods in terms of prediction performance, as explained previously. 80.52% accuracy obtained using real-time data with our proposed method is a similar accuracy with the models developed using historical data in the literature. Obtaining similar accuracy with real-time data caused us to evaluate our model as successful. In addition, the study proposes a semi-autonomous system and leaves the final choice to the decision makers. For this reason, it is predicted that model error rates close to 20% can be corrected by decision makers. Our model will automate the pre-assignment tasks that require a lot of work, thus allowing decision makers to exert much less effort.

Table 6. Accuracy and AUC scores of machine learning algorithms on Stop-word cleaning applied Dataset

Method	Accuracy				AUC			
	TF	TFIDF	W2V	D2V	TF	TFIDF	W2V	D2V
LR	0.7648	0.7933	0.7791	0.734	0.959	0.9717	0.9713	0.9573
SVM	0.7316	0.7933	0.8052	0.7055	0.959	0.9735	0.9749	0.9549
KNN	0.2375	0.6627	0.7173	0.4537	0.6613	0.945	0.9441	0.8313
RF	0.7221	0.7221	0.7791	0.7245	0.9529	0.9514	0.966	0.9437
XGB	0.0974	0.5796	0.7791	0.4608	0.5	0.8973	0.9701	0.8655

The results of the data set containing normalization and stemming preprocessing are shown in tables 7 and 8, respectively. There were no significant changes in the performance of the machine learning algorithms for either. However, some results show extremely low performance, such as the 38% accuracy of KNN in the stemmed dataset or the 31% accuracy of XGB in the normalized dataset. These low accuracies can be explained by overfitting due to the large hyperparameter optimization space of the algorithms.

Table 7. Accuracy and AUC scores of machine learning algorithms on normalization process applied Dataset

Method	Accuracy				AUC			
	TF	TFIDF	W2V	D2V	TF	TFIDF	W2V	D2V
LR	0.772	0.7838	0.7363	0.6746	0.9626	0.9718	0.9627	0.9436
SVM	0.7411	0.7815	0.7862	0.6437	0.9604	0.9751	0.9657	0.9263
KNN	0.1591	0.677	0.6722	0.5701	0.8601	0.9489	0.9456	0.9171
RF	0.7245	0.7292	0.772	0.6698	0.9541	0.9529	0.9673	0.942
XGB	0.3135	0.5202	0.7221	0.6461	0.7071	0.86	0.9638	0.9362

Table 8. Accuracy and AUC scores of machine learning algorithms stemming process applied Dataset

Method	Accuracy				AUC			
	TF	TFIDF	W2V	D2V	TF	TFIDF	W2V	D2V
LR	0.7458	0.7815	0.7292	0.7197	0.9519	0.971	0.9587	0.9513
SVM	0.7458	0.791	0.7672	0.696	0.9555	0.974	0.9665	0.9397
KNN	0.38	0.6556	0.7126	0.6057	0.7539	0.9463	0.9234	0.9008
RF	0.7672	0.7316	0.7815	0.715	0.9673	0.9642	0.9616	0.9477
XGB	0.5653	0.5629	0.7387	0.715	0.8912	0.9231	0.9609	0.9577

FASTText, another algorithm applied to our dataset, failed to outperform the other machine learning methods, achieving a low prediction accuracy of 67%. The algorithm splits each word within the sample text based on a given n-gran size for representation; therefore, the word vectors may have lost their discriminative power with respect to the field labels.

For example, the FASTText representation of the word "detay" contains $<$ "de","et","ta","ay" $>$ sub-word vectors when the n-gram size is set to 2. But the word "ay" also has many different meanings in Turkish. Therefore, it may be possible that contribution of sub-words to representation has some negative effect in Turkish language.

The Receiver Operating Characteristic (ROC) curves for each class obtained from the SVM-Word2Vec-NOSW model, which performed the best among all models, are shown in Figure 7. The ROC curve is one of the performance measurement methods that shows how well the model can separate classes for a binary problem. The ROC curve is plotted with the TPR (True Positive Rate) against the FPR (False Positive Rate), with the TPR on

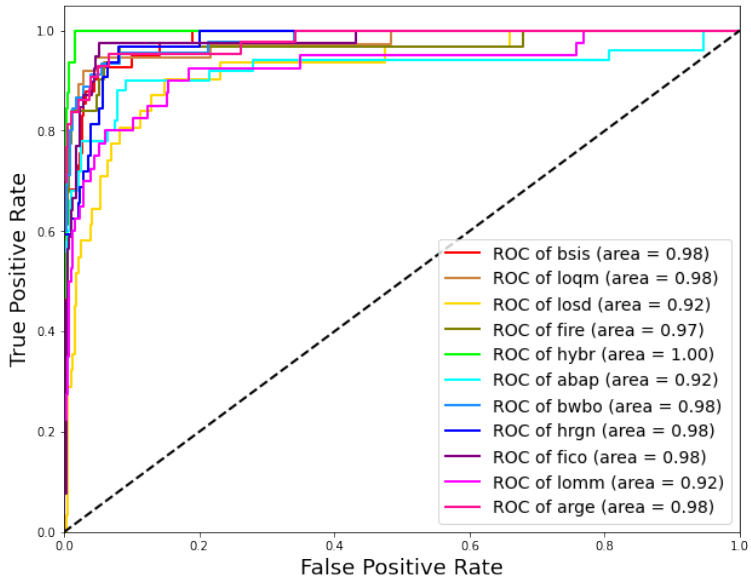

Fig. 7. Plot of Area Under the Curve (AUC) for each class

the y-axis and the FPR on the x-axis. If the AUC value is equal to 1, this indicates that the model is excellent at discriminating between the class labels of the samples. Since the dataset used in our study consists of 11 classes, the OvR (One-versus-Rest) approach was used, where each class was left alone to label the problem as binary to plot the ROC curves.

After predicting the class vectors of the new task text, the recommendation process is started. Our system was developed as a microservice that can be integrated into a company's portal or ERP system. The front-end design of our application is shown in Figure 8. When a task is created and the information fields such as subject, description, project and deadline are filled in, the automatic system is triggered to recommend the most suitable candidates for the task. Although the machine learning model predicted the text as "LoSd" class, our model recommended some employees who have skills for "LoQm" first. The reason for this result is that our machine learning model does not predict a specific class, but a vector of class distribution. It can be interpreted that the vector of task input might have included some different requests for "LoSd" besides "LoQm", and therefore employees who have "LoQm" skills are ranked first because the decision support system recommends the most suitable candidate according to cosine similarity.

5. Conclusion

Efficiently managing multiple projects and ensuring timely completion poses significant challenges for IT and consulting firms. It is necessary to assign tasks to the right people in order to complete the tasks at hand together with limited human resources. The assignment of tasks can only be decided by senior staff such as project managers or RD

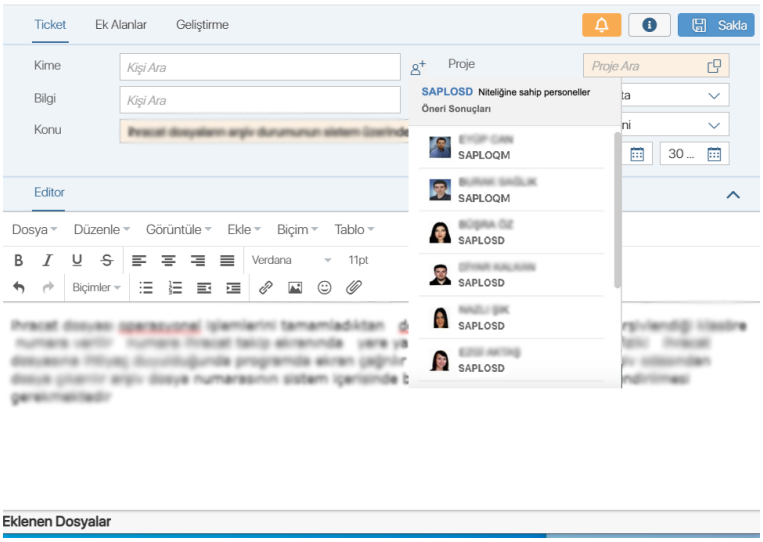

Fig. 8. User interface of proposed recommendation system

managers who know the qualifications of the staff. However, managing and controlling task assignment is somewhat tedious, especially in large organizations that have numerous teams and employees with different skills and qualifications.

Implementing an automated system that processes tasks and quickly recommend the most suitable employees has the advantage of shortening project run times and avoiding potential assignment problems. Our work involves the development of a decision support service to recommend the most appropriate employees by analyzing the task text using various methods to facilitate task assignment.

When a new task is entered through the system, the vector containing the task requirements is automatically predicted and employees are recommended based on skill similarities. Once the given tasks are completed, the qualifications of the assigned personnel are automatically updated with the task vector, using the same system. In this way, tracking the progress of personnel is simplified and does not require further human intervention. Since each employee is familiar with the required tasks and the qualifications are updated regularly, the distribution of qualified personnel within the company can be systematically and statistically recorded to support the HR department.

Establishing an automatic system, which handles tasks and quickly recommend the most suitable employees, provide advantages on shorten completion times of project and might have prevent potential assignment issues. Our work includes the development of the decision support service, which aims to recommend the most suitable personnel by analyzing the task text with various methods in order to facilitate task assignments. Unlike other studies in the literature, our system developed to integrate a living real-time system rather than the perform an optimization on static task/personnel dataset. When a new task is entered over the system, the vector which includes requirement of task is automatically predicted and employee are recommended based on qualification similarities. Just after the related tasks is completed, qualifications of assigned personnel is updated with task

vector automatically using the same system. Thus, the tracking of personnel progress is simplified and not need more human interference. Besides, since the each employee get experienced with skill required tasks and qualifications are updated regularly, the qualified staff distribution within the company can reported in systematically and statistically to use in process of human resource department.

Since the main goal of our study is to present the general architecture of a decision support system working with integrated enterprise software, some new generation transformer-based text representation approaches such as ELMo [26], BERT [10] have not yet been considered. In future studies, the success rate of the recommender system will be improved by adding new class labels for subdepartments and using state-of-the-art text representation models. In addition, although employee workload is recorded in the portal system, this information is not functionally used in task assignment. In the next version of the application, a more complex system will be developed that includes workload, qualifications, and task assignments.

References

1. SAP Global | Platin İş Ortağı - Detaysoft, https://detaysoft.com/tr-TR/index
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2623–2631 (2019)
3. Al-Hawari, F., Barham, H.: A machine learning based help desk system for IT service management. Journal of King Saud University-Computer and Information Sciences 33(6), 702–718 (2021), iSBN: 1319-1578 Publisher: Elsevier
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Tech. Rep. arXiv:1607.04606, arXiv (Jun 2017), http://arxiv.org/abs/1607.04606, arXiv:1607.04606 [cs] type: article
5. Cai, X., Li, K.N.: A genetic algorithm for scheduling staff of mixed skills under multi-criteria. European Journal of Operational Research 125(2), 359–369 (2000), iSBN: 0377-2217 Publisher: Elsevier
6. Carki, K., Geutner, P., Schultz, T.: Turkish LVCSR: towards better speech recognition for agglutinative languages. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100). vol. 3, pp. 1563–1566 vol.3 (Jun 2000), iSSN: 1520-6149
7. Charmaz, K., Belgrave, L.L.: Thinking about data with grounded theory. Qualitative Inquiry 25(8), 743–753 (2019), iSBN: 1077-8004 Publisher: SAGE Publications Sage CA: Los Angeles, CA
8. Chen, R., Liang, C., Gu, D., Leung, J.Y.: A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution. International Journal of Production Research 55(21), 6207–6234 (2017), iSBN: 0020-7543 Publisher: Taylor & Francis
9. Cheng, H., Chu, X.: Task assignment with multiskilled employees and multiple modes for product development projects. The International Journal of Advanced Manufacturing Technology 61(1), 391–403 (2012), iSBN: 1433-3015 Publisher: Springer
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Tech. Rep. arXiv:1810.04805, arXiv (May 2019), http://arxiv.org/abs/1810.04805, arXiv:1810.04805 [cs] type: article
11. Dooley, L., Lupton, G., O'Sullivan, D.: Multiple project management: a modern competitive necessity. Journal of Manufacturing Technology Management 16(5), 466–482 (Jan 2005),

https://doi.org/10.1108/17410380510600464, publisher: Emerald Group Publishing Limited
12. Eryiğit, G.: ITU Turkish NLP web service. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 1–4 (2014)
13. Garcia, I., Pacheco, C., Arcilla-Cobián, M., Calvo-Manzano, J.: Mypmp: A plug-in for implementing the metamodeling approach for project management in small-sized software enterprises. Computer Science and Information Systems 13(3), 827–847 (2016)
14. Guo, L., Vargo, C.J., Pan, Z., Ding, W., Ishwar, P.: Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling. Journalism & Mass Communication Quarterly 93(2), 332–359 (2016)
15. Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of operational research 207(1), 1–14 (2010), iSBN: 0377-2217 Publisher: Elsevier
16. Kane, H., Tissier, A.: A Resources Allocation Model for Multi-Project Management. In: 9th International Conference on Modeling, Optimization & Simulation (2012)
17. Lagesse, B.: A Game-Theoretical model for task assignment in project management. In: 2006 IEEE International Conference on Management of Innovation and Technology. vol. 2, pp. 678–680. IEEE (2006)
18. Lahitani, A.R., Permanasari, A.E., Setiawan, N.A.: Cosine similarity to determine similarity measure: Study case in online essay assessment. In: 2016 4th International Conference on Cyber and IT Service Management. pp. 1–6 (Apr 2016)
19. Le, Q.V., Mikolov, T.: Distributed Representations of Sentences and Documents. Tech. Rep. arXiv:1405.4053, arXiv (May 2014), http://arxiv.org/abs/1405.4053, arXiv:1405.4053 [cs] type: article
20. Li, X., Nie, M., Yang, G., Wang, X.: The study of multi-project resource management method suitable for research institutes from application perspective. Procedia Engineering 174, 155–160 (2017)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
22. Mitkov, R.: The Oxford handbook of computational linguistics. Oxford University Press (2004)
23. Mo, Y., Zhao, D., Du, J., Syal, M., Aziz, A., Li, H.: Automated staff assignment for building maintenance using natural language processing. Automation in Construction 113, 103150 (2020), iSBN: 0926-5805 Publisher: Elsevier
24. Möhring, R.H.: Minimizing costs of resource requirements in project networks subject to a fixed completion time. Operations Research 32(1), 89–120 (1984), iSBN: 0030-364X Publisher: INFORMS
25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.: Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12, 2825–2830 (2011), iSBN: 1532-4435 Publisher: JMLR. org
26. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. Tech. Rep. arXiv:1802.05365, arXiv (Mar 2018), http://arxiv.org/abs/1802.05365, arXiv:1802.05365 [cs] type: article
27. Ponsteen, A., Kusters, R.J.: Classification of human-and automated resource allocation approaches in multi-project management. Procedia-Social and Behavioral Sciences 194, 165–173 (2015), iSBN: 1877-0428 Publisher: Elsevier
28. Rehurek, R., Sojka, P.: Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3(2), 2 (2011)
29. Shahariar, G.M., Biswas, S., Omar, F., Shah, F.M., Hassan, S.B.: Spam review detection using deep learning. In: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp. 0027–0033. IEEE (2019)

30. Vijayarani, S., Ilamathi, M.J., Nithya, M., et al.: Preprocessing techniques for text mining-an overview. International Journal of Computer Science & Communication Networks 5(1), 7–16 (2015)
31. Vogrinčič, S., Bosnić, Z.: Ontology-based multi-label classification of economic articles. Computer Science and Information Systems 8(1), 101–119 (2011)
32. Vyas, M.J., Bhanderi, S.D.: A Survey on Pre-processing Techniques for Text Mining. Data Mining and Knowledge Engineering 6(2) (2014), http://www.ciitresearch.org/dl/index.php/dmke/article/view/DMKE022014006, number: 2

Halil Arslan received BS, MS and, PhD, degree in electronic and computer education from Sakarya University, Turkey in 2004-2015 respectively. Since 2017, he has been teaching operation systems, cyber security and, computer networks courses as an assistant professor in the Computer Engineering Department at University of Sivas Cumhuriyet. His research interests are computer networks, cyber security and, software engineering.

Yunus Emre Işik received his bachelor's and master's degrees in Management Information Systems from Mehmet Akif Ersoy University and Cumhuriyet University, respectively. He is pursuing his Ph.D. in Electrical and Computer Engineering at Abdullah Gul University. He is currently working as a research assistant in the Department of Management Information Systems at Cumhuriyet University in Sivas, Turkey. His research focuses on the implementation of AI models in various domains such as text mining, image processing and bioinformatics.

Yasin Görmez received the graduate degree from Computer Engineering Department, Meliksah University, the MSc degrees with high honor from the Electrical and Computer Engineering Department, Abdullah Gul University, and Ph. D. degrees with high honor from the Electrical and Computer Engineering Department, Abdullah Gul University, in 2015, 2017 and 2022 respectively. He served as a research assistant between 2015 and 2023 in management information systems department of Sivas Cumhuriyet University, Sivas, Turkey. Now, he is an assistant professor in management information systems department of Sivas Cumhuriyet University, Sivas, Turkey. He has particularly honed his skills in designing deep learning methods and has developed deep learning techniques in various fields such as bioinformatics, natural language processing, image processing, and cybersecurity

Mustafa Temiz graduated from Computer Engineering Department of Erciyes University. He received the master's degree in management information systems from Sivas Cumhuriyet University. Currently a Ph.D. student in Electrical and Computer Engineering at the Abdullah Gul University and a research assistant in Department of Management Information Systems, Cumhuriyet University, Sivas, Turkey.

Received: September 22, 2023; Accepted: October 11, 2023.