HU ISSN 1785-1270

PRODUCTION SYSTEMS AND INFORMATION ENGINEERING

A Publication of the University of Miskolc

VOLUME 9 (2020)

MISKOLC UNIVERSITY PRESS

HU ISSN 1785-1270

PRODUCTION SYSTEMS AND INFORMATION ENGINEERING

A Publication of the University of Miskolc

VOLUME 9 (2020)

MISKOLC UNIVERSITY PRESS

University of Miskolc
Responsible for publication:
Responsible for layout and editing: Dr. Attila Károly Varga
Published by the Miskolc University Press
under the leadership of Attila Szendi
Responsible for duplication: Erzsébet Pásztor
Number of copies printed:
Put to the Press on
Number of permission:
HU ISSN 1785-1270

EDITORAL BOARD

Production Systems and Information Engineering
Volume 9 (2020), pp. 5-18

EFFICIENCY ANALYSIS OF NNS METHODS

Balázs Bolyki
University of Miskolc, Hungary
Student of Computer Science Engineering
bolyki@iit.uni-miskolc.hu

Dávid Polonkai
University of Miskolc, Hungary
Student of Computer Science Engineering
polonkai3@iit.uni-miskolc.hu

Dr. László Kovács
University of Miskolc, Hungary
Department of Information Technology
kovacs@iit.uni-miskolc.hu

[Received . . . and accepted . . .]

Abstract. Nearest Neighbor Search is a key operation in multiple information technologies fields, for example, string matching, plagiarism detection, natural language processing, image clustering, etc. It is crucial, that we have cost efficient methods and structures for retrieving data based on similarity. We conducted a survey of two popular – Vantage Point tree and Locality Sensitive hashing –, and one more recent – Prefix tree with clustering – NNS methods. In order to perform a wide range of tests on these algorithms, we adapted each to Python language, and developed multiple tests. In this paper we present the description of the three algorithms and the results of our tests. We aim to provide an informative comparison of the three major Nearest Neighbor Search structures.

Keywords: NNS, VP-tree, LSH, Prefix tree, Nearest Neighbor, search, comparison

1. Introduction

The search for similar objects is a key operation in general information management. When we retrieve information based on the high level of similarity – or in other words, the smallest distance – between known and unknown data, we execute Nearest Neighbor Search (NNS). The applications of NNS are numerous and also are the methods invented to make it faster, more effective. The query for nearest neighbor elements is used, among others in autocomplete and spell checking [1], plagiarism detection [2, 3, 4, 5, 6], natural language processing [7], image clustering [8], and medical databases [9]. The time efficiency of NNS operations is a crucial cost factor in the whole information system. Similarity and distance are two measures approaching the same problem from the opposite directions. The more similar two objects are to each other, the less the distance is between them, while the greater the distance between them, the less similar they are. Therefore we use both measures in this essay. This work focuses on an important application area, the search in a word repository. This paper's main purpose is twofold, first to analyze the cost efficiency of the known NNS methods and to adapt them to the investigated problem domain, namely the similarity search in word repositories. The work analyzes two popular methods, VP- tree and LSH algorithms and compares them with a recent approach, the prefix-tree NNS structure. The performed tests cover both time efficiency and search accuracy analyses. Our goal is to list, explain and evaluate these methods and to give the readers an insight into the strengths and weaknesses of each.

Nearest Neighbor Search. Nearest Neighbor Search (NNS) is the operation, which retrieves a data object, from a given dataset, based on its distance from a query object. Formally, if $\mathbb{U}$ is the universe and $\mathbb{S} \subseteq \mathbb{U}$ is the dataset, in which we search, and $q \in \mathbb{U}$ is the query object, we can define NNS as follows.

$$d : \mathbb{U}^2 \to \{x \in \mathbb{R} \mid x \geq 0\} \tag{1.1}$$

$$\begin{aligned} N_1(q) = \{y \in \mathbb{S} \mid \forall z \in \mathbb{S}, d\,(q, y) \leq d\,(q, z)\} \\ |N_1(q)| = 1 \end{aligned} \tag{1.2}$$

$$\begin{aligned} N_k(q) = \{\forall x \in N_k(q), y \in \mathbb{S} : d(q, y) < d(q, x) \Rightarrow y \in N_k(q)\} \\ \text{and } |N_k(q)| = k \\ \text{where } k \in \mathbb{N}^+ \text{ and } k \leq |\mathbb{S}| \end{aligned} \tag{1.3}$$

Formula 1.1 defines a distance function d, which we use in the Nearest Neighbor Search definition in the Formula 1.2. In Formula 1.3 we also define the generalization of NNS to k-NNS, where we retrieve the k nearest neigbors of the query q, from the dataset $\mathbb{S}$.

2. Description of evaluated algorithms

The three evaluated approaches, Vantage Point tree (VP-tree), Locality Sensitive Hashing, and Prefix tree with clustering (prefix tree) each realize an efficient way to execute nearest neighbor search operation, however, their

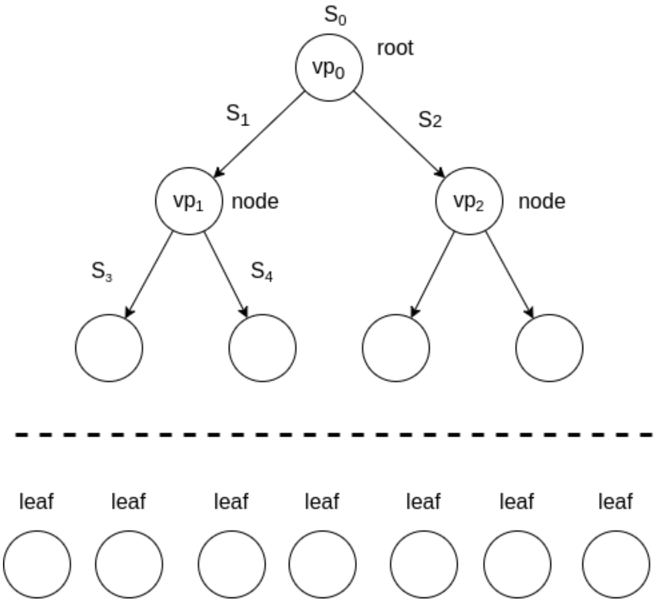


Figure 1. Representation of VP-tree concept

approach is quite different. In this section we will describe the approach of each algorithm generally.

2.1. Vantage Point tree

VP-tree realizes NNS operation based on the concept of general metric spaces. It requires the dataset containing the data objects and the distance function d to build the indexing structure. Here, the distance function must satisfy the properties of a metric (positivity, identity of the same objects, symmetry, and triangle inequality), because the VP-tree can use the triangle inequality to prune branches, when searching.

Our main reference for the VP-tree is the paper [10]. The main concept of the VP-tree algorithm to solve the NNS problem is to build an indexing structure in such a way that the data objects get distributed according to their distance from vantage points, which are chosen from the dataset, one at each non-leaf node. At the building stage if the current dataset S_i does not fit into a leaf, we apply Formula 2.1, where we denoted the base dataset as S_0 and the created subsets are S_1, S_2 and S_i in further nodes, while m is the median of the distances from the vantage point (the distance between the vantage point and the elements of S_i is calculated for each element during the partitioning at each node). This partitioning scheme would be recursively repeated at each node, until all created subsets are allocated into leaves.

$$\begin{aligned} S_1 &= \{s \in S_0 \mid d(s, vp_0) < m\} \\ S_2 &= \{s \in S_0 \mid d(s, vp_0) \geq m\} \end{aligned} \tag{2.1}$$

The algorithm for searching in the tree is presented in the paper [10]. It uses a recursive strategy to traverse the tree and prunes branches of it based on the triangle inequality.

2.2. Locality Sensitive Hashing

The Locality Sensitive Hashing is based on hash functions. The hash functions map a key into a hash value. The hash function represents the same value for the same input. It is possible, that two different keys mapped into the same hash value. This is hash collision. [11]
The LSH uses hash collision to find similar keys. This method hashes data objects by multiple hash functions and stores the hash value and key pairs. Therefore if more collisions are found in different hash functions, it is more probable that keys are similar. To find similarities the algorithm hashes the query point and returns the elements from the buckets that contain that point. [12] This method also uses minhash. It is a special hash function, that we execute multiple times to receive the same hash values for similar keys. In our case it is used to map the keys into hash values, which we add to the LSH function. For better understanding we have to declare shingling, which is a method, that can divide its input into k lenght sequences. After shingling the hash value, the method puts them into the buckets of the LSH function. The idea of the LSH algorithm is represented in Figure 2. In our case the algorithm we use is a library based on the [13].

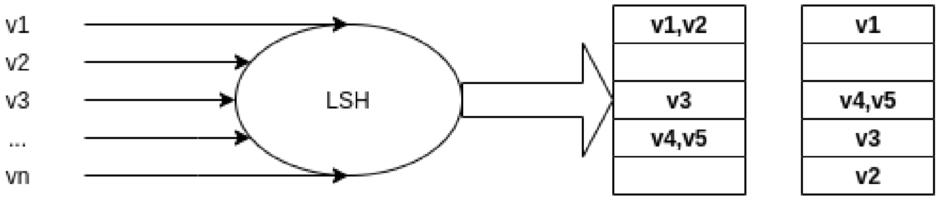


Figure 2. Representation of Locality Sensitive Hashing concept

2.3. Prefix-tree with clustering

This method uses clustering for preprocessing and the prefix-tree for storing the actual words. The prefix-tree is a special kind of tree, in which the nodes are letters. A word is represented by the path of letters.

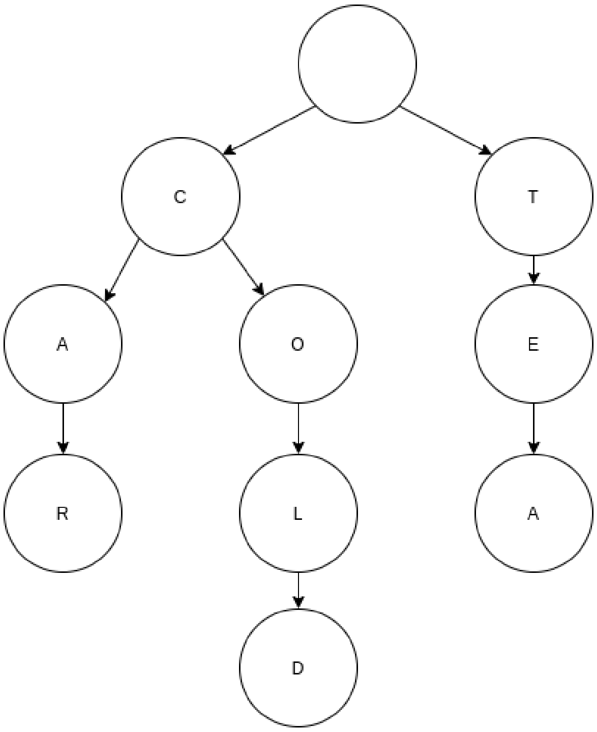


Figure 3. Representation of Prefix-tree concept

Prefix-tree. In case of insertion the method starts to go down in each branch and finds the current letter in the structure. If there is no such a letter, it creates a new node. So for example if we want to find the *word*, then in the first level from the root we look for a letter *w*. In the query for 1NN search we go down in each branch and check if the current character is the same in the structure. If they are different, it means that the distance between them is one more. By this method we can find the best fitting similarity in the structure, but if we go down in every branch it costs much in time, so the method also uses a limit parameter, which determines the maximum distance in a branch. If the distance reaches this value, we cut the branch.

Prefix-tree with clustering. This method also uses clustering. This means that we map the words into different sets. These sets are further from each other and does not overlap. In each cluster there is a tree and the words are stored there. The stucture building starts with calculating the central points of the clusters. After this we build all the prefix trees. In case of insertion the algorithm calculates the nearest cluster and builds the word into the tree. For calculation the method represent the words into vectors and runs a minimum search on the cluster distance from the word in order to find the nearest cluster. The query method works the same, so it finds the cluster, then goes down in the tree and searches for the word as described above. The base implementation of the method is presented in [14].

3. Adaptation of investigated methods

In order to test the methods, we adapted them to a common implementation framework. We chose the Python programming language, because it provides libraries and partial implementation of the used methods, and it hides most of the lower level operations, so we could focus on the unique parts of the algorithms.
The parameters of the computer in which the tests have run are the following:

- cpu: Intel®Core™i7-8550U @ 1.80GHz
- ram: 7.7 GiB @2400 MHz and 7.5 GiB of swap memory
- swap device: M.2 Solid state drive with 600 MB/s data transfer rate
- operating system: Ubuntu 18.04.3 LTS 64-bit

3.1. Vantage Point tree

Our VP-tree implementation is based on the paper [10]. We should also reference [15] for the implementation of the Autosorting list and as a structural example. However, due to our different platforms and emphasis – namely us using mainly Python language and our tests focusing on speed rather than page access rates – we diverged from [10] in multiple aspects. We followed the proposition of the paper for

- building the tree and
- executing k-NNS in the tree.

We performed the following extensions of the base model:

- We set the leaf size based on the quantity and not the size of the data objects, that would be contained within. This is more advantagous in a higher level programming language, since it provides more information, than does the data object size.
- We designed our own insertion algorithm. This looks up the leaf, in which the data object to be inserted should be placed, and allocates it

in the leaf, it still has space. If does not have space, then the algorithm starts to go up the tree, and checks if the subtree marked my the checked node (its descendants are its subtree) has space. If it has, the algorithm retrieves the data stored in that subtree, then rebuilds it. If the entire tree is full, then the entire tree is rebuilt.

3.2. Locality Sensitive Hashing

The algorithm originally used for similarity search in big texts like paragraphs. So we have to tweak the parameters to work with word similarity search. One of the most important parameters is the `permutations`, which determines the number of permutations by the minhash function. If this parameter is big, then the memory consumpsion and building time of this method increase, but the query time will be less. We determined that the ideal number for this parameter is 60. The other parameter is `n_gram`, which determines the shingle size for minhash. We choose this to be 2, because by this we can enter bigger than 2 lenght words. To make sure that the words contain more than 2 characters we built a function, which fills the words, which are less than 4 characters with _ characters. Later in the query we also run this on the searched word, to make sure that the right results are returned. We also changed the `no_of_bands` parameter, which defines the number to break the minhash signature before hashing into the buckets. The accuracy is affected by this parameter. We set the sensitivity to 2 which means the number of buckets texts must share to be declared as similar. Moreover we implemented the k-NNS methods, which calculates the k nearest from the result set.

3.3. Prefix-tree with clustering

In case of this method we also modified the parameters. The most important parameter for this method is the `limit`, which determines the maximum distance that we want to search in the tree. As is mentioned this parameter determines how deep the search can go down to find the best match in a branch. The parameter has to be larger than the distance of the searched word and its nearest neighbour, otherwise the algorithm does not return a word. But if the `limit` is much larger than the nearest neighbor, then the search time increases dramatically. In most of our tests we declared the limit parameter as 3, in some of them we used also 3 and 6. The other crutial parameter is the number of clusters. Because the word cluster determination is a step with high costs we have to choose it properly. our wordlists use 50000 to 2.4 million words, therefore we choose a rather big clusternumber, which is 100. In addition to the parameter settings we created methods for k-NNS search and for the insertion. The k-NNS search method goes through the tree until k number of results are found or the limit distance reached.

4. Performed tests

To test the algorithms on similar word searching, we have to create wordlists. These wordlists contain Hungarian and English words. The building wordlist consists of 2.4 million Hungarian words. In order to build structures with less words we take a sublist from this. The searchlist is a Hungarian and mostly English wordlist. These words are not in the building list.
We created several tests:

- build time tests for different wordlists,
- search time tests for different wordlists and different percentage of known-unknown words,
- k-NNS tests: scaling depending on the k value;
- accuracy tests,
- insert time tests.

Most of the tests measure time required to do the task, except for accuracy tests, which measure that the returned word is the real nearest neighbour.

Build test. The building time test is, where we tested the required time to build up the structure from zero. We ran this test on different wordlists, from 200000 to 2.4 million in each cycle we increased the number of words by 200000. The results of the test is represented in Figure 4.

Accuracy comparison. We measured the accuracy of each algorithm by first building a structure from a wordlist of 50000 words, then executing nearest neighbor search in two ways. We retrieved the nearest neighbor using the evaluated algorithm, and also using a linear search algorithm, that we wrote for this purpose. We accepted the result returned by the linear search algorithm as the real nearest neighbor, and compared it to the result returned by the evaluated structure. If the two returned objects were of the same distance from the query point, then we considered the search accurate, otherwise we considered it inaccurate. We looked up the nearest neighbor of 20 words, and calculated the percentage of accuracy. We also executed this test for different proportions of known-unknown words in the wordlist (for example, 20% of the search list was also in the structre). The results are presented in Figure 5.

Search test. The search time tests measure the time required to run the query method. We used 20 words size wordlists, and measured the time each algorithm takes to look up the nearest neighbor of the 20 words, then divided the result by 20 to receive the average search time per word value for each algorithm. The results for the 20 unknown words (words not inside the structure) are represented in Figure 6.

K-NNS test. We tested the algorithms for their scaling given multiple k values, when we execute k-NNS operation. To do so, we built the structures from 50000, 100000, . . . , 400000 words size wordlists, then executed k-NNS, with the above mentioned 20 words search list (none of the 20 words were inside the structure), with the k values of 1, 3, 10. As before, we present the average search time per query word in Figure 7.

Insert time test. The insert time test is a test, in which we measure the time needed to insert 100 words into the different structures. We executed this insertion on structures built from 200000 to 1.2 million words size wordlists. The results of the test are represented in Figure 8.

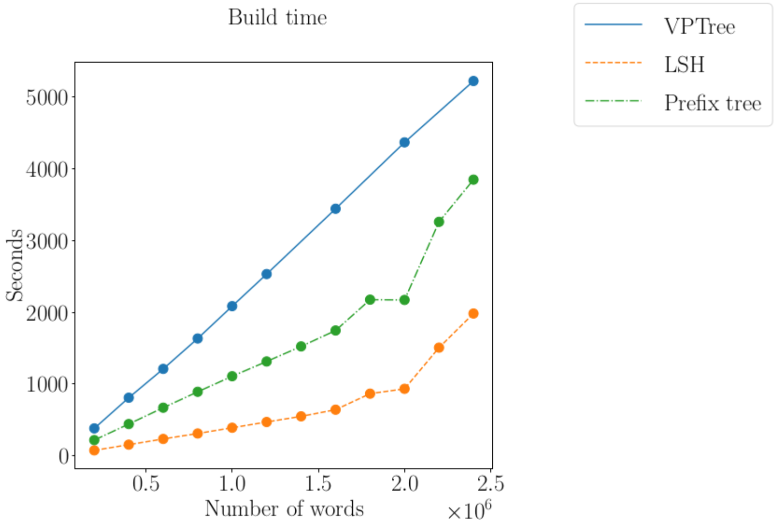


Figure 4. Building time comparison

5. Conclusion

In this article we have listed a few application areas of Nearest Neighbor Search, described a few import Nearest Neighbor Search algorithms, namely:

- Vantage Point tree (VP-tree)
- Locality Sensitive hashing (LSH)
- Prefix tree with clusterng (Prefix-tree)

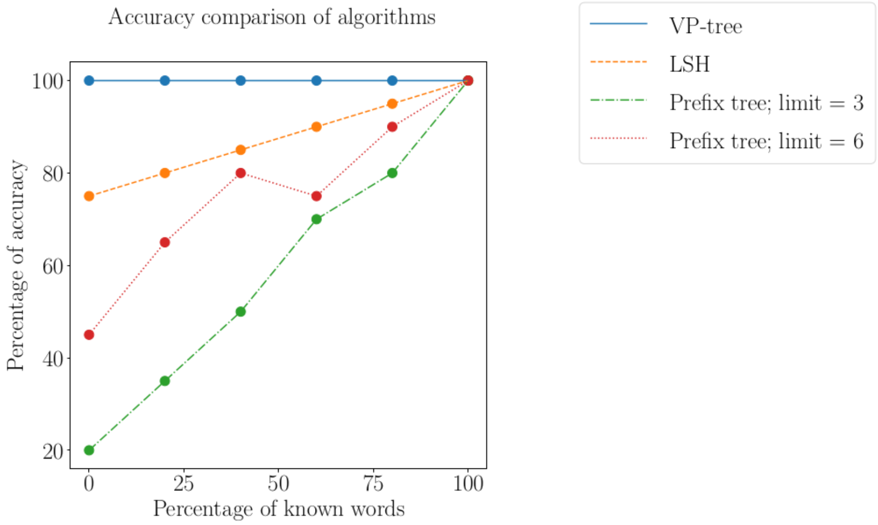


Figure 5. Accuracy comparison

VP-tree and LSH are more conventional and widely known methods, while Prefix-tree is a more recent approach. We adapted each algorithm to a uniform framework to perform a range of tests. From the results of our tests we can conclude the following:

- LSH takes the less time to build up.
- VP-tree is the most accurate of the three methods.
- Prefix-tree and LSH clearly outperform VP-tree in regards of search time.
- The speed comparison between Prefix-tree and LSH is not one-sided, but Prefix-tree scales better, while LSH is less affected by whether the searched word is inside the structure or not.
- Each algorithm scales reasonably depending on the k value at k-NNS operation.
- The difference between insertion speed of the three algorithms is rather small, Prefix-tree being slightly slower on the evaluated interval.

6. Acknowledgements

The described article was carried out as part of the EFOP-3.6.1-16-2016-00011 “Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialisation” project implemented in the framework of the Szechenyi 2020

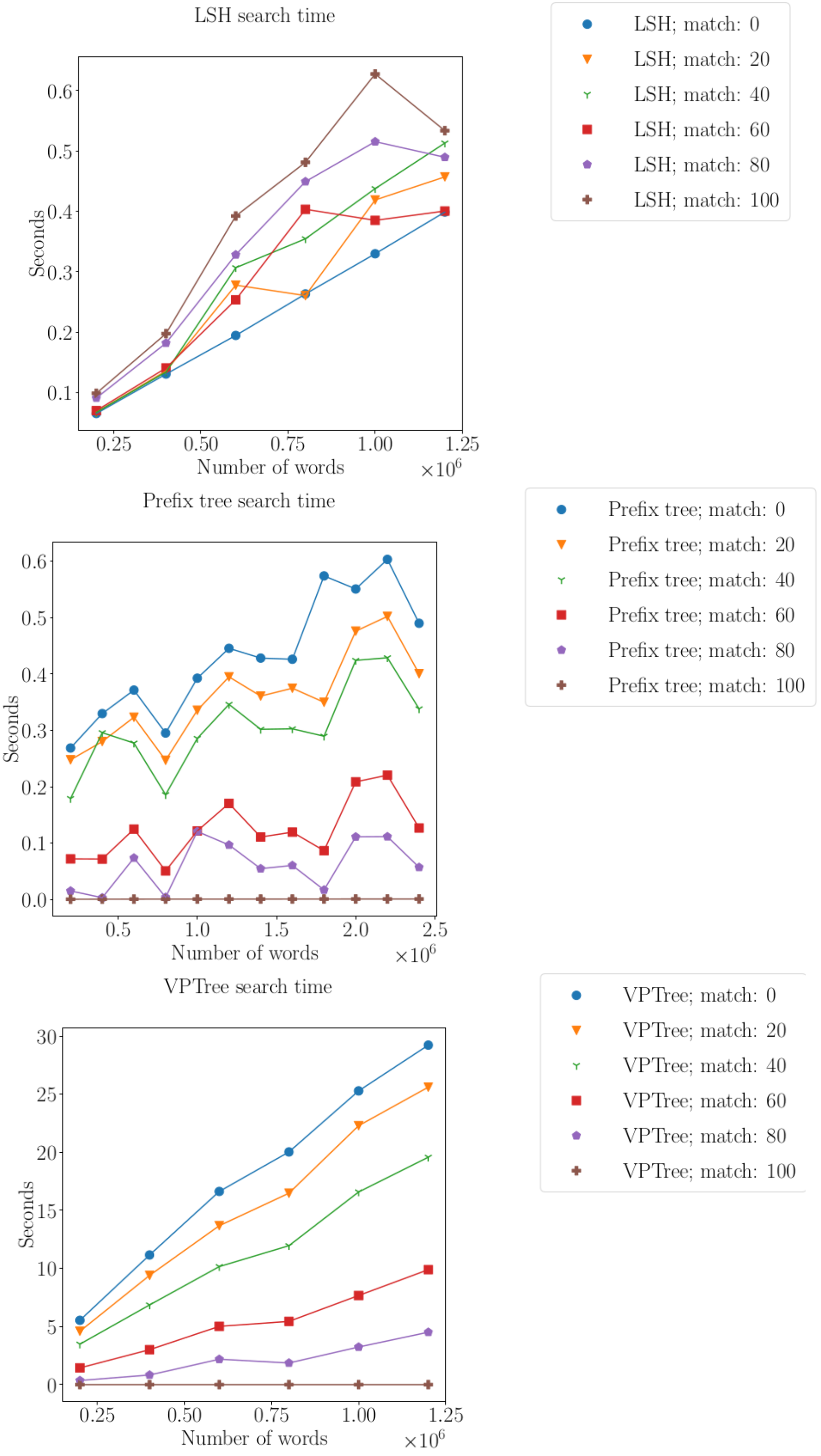


Figure 6. KNN search time comparison

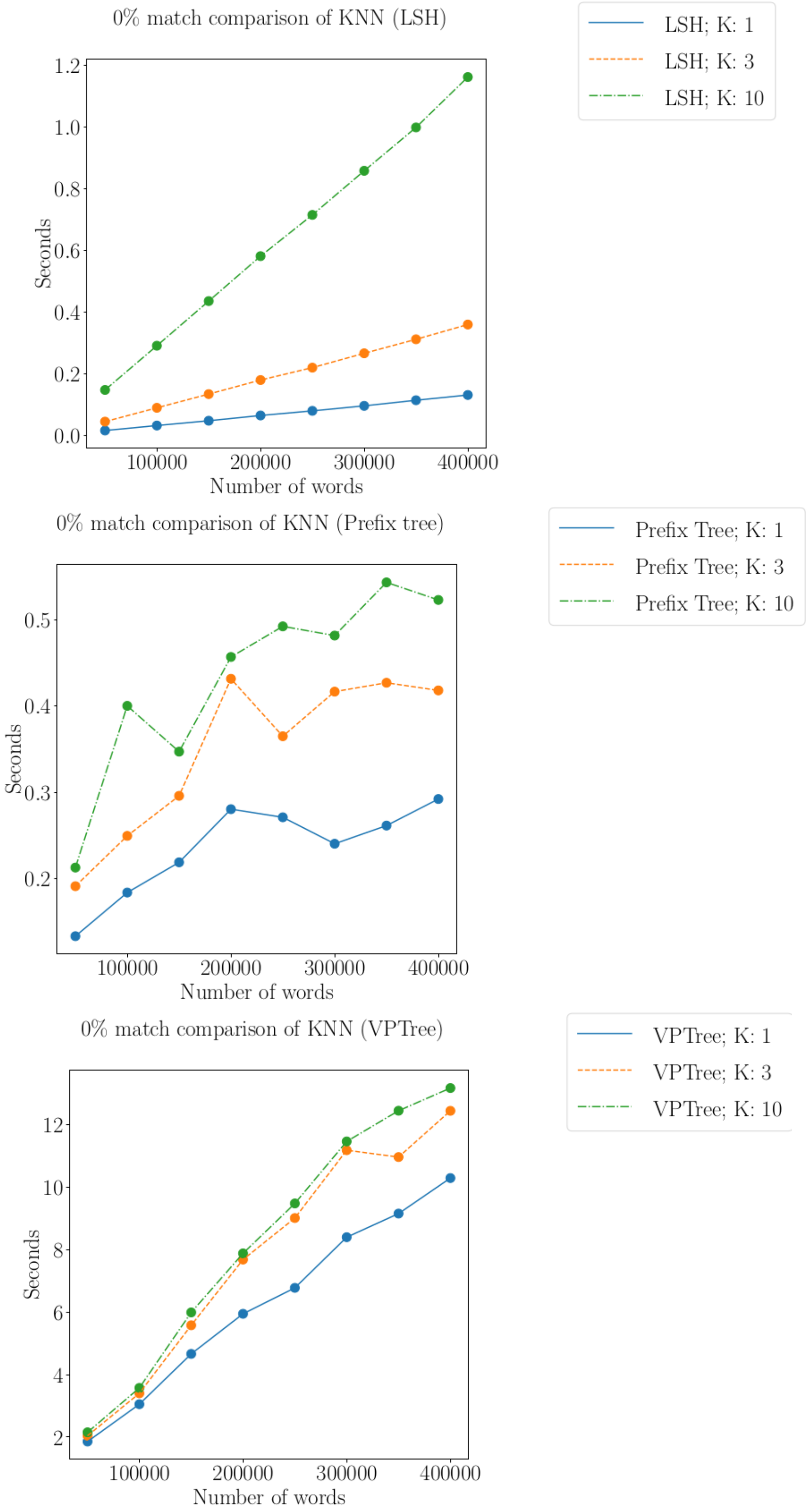


Figure 7. KNN search time comparison

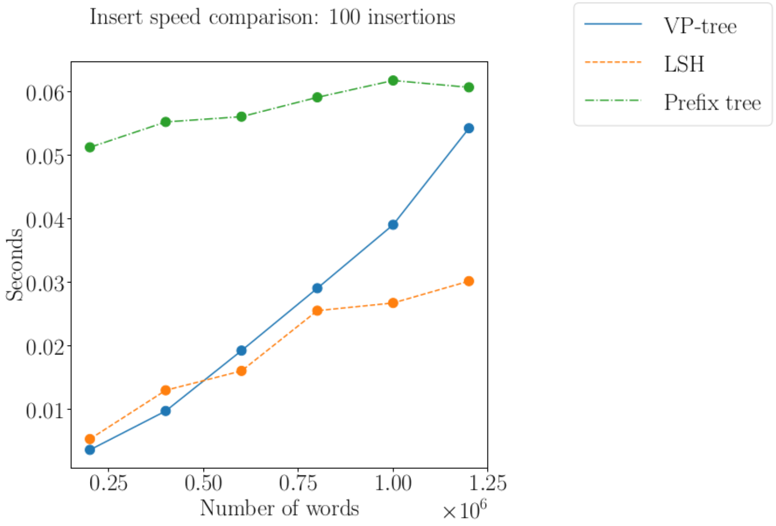


Figure 8. Insert comparison

program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] Yulianto, M. M., Arifudin, R., and Alamsyah, A.: Autocomplete and spell checking levenshtein distance algorithm to getting text suggest error data searching in library. *Scientific Journal of Informatics*, **5**(1), (2018), 75.

[2] Agrawal, M. and Sharma, D. K.: A state of art on source code plagiarism detection. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, 2016, pp. 236–241.

[3] Baba, K.: Fast plagiarism detection using approximate string matching and vector representation of words. In *Behavior Engineering and Applications*, pp. 67–79, Springer, 2018.

[4] Srivastava, S., Mukherjee, P., and Lall, B.: implag: Detecting image plagiarism using hierarchical near duplicate retrieval. In *2015 Annual IEEE India Conference (INDICON)*, IEEE, 2015, pp. 1–6.

[5] Potharaju, R., Newell, A., Nita-Rotaru, C., and Zhang, X.: Plagiarizing smartphone applications: attack strategies and defense techniques. In *International symposium on engineering secure software and systems*, Springer, 2012, pp. 106–120.

[6] Hussain, S. F. and Suryani, A.: On retrieving intelligently plagiarized documents using semantic similarity. *Engineering Applications of Artificial Intelligence*, **45**, (2015), 246–258.

[7] Goyal, A., Daumé III, H., and Guerra, R.: Fast large-scale approximate graph construction for nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1069–1080.

[8] Liu, T., Rosenberg, C., and Rowley, H. A.: Clustering billions of images with large scale nearest neighbor search. In *2007 IEEE workshop on applications of computer vision (WACV'07)*, IEEE, 2007, pp. 28–28.

[9] Korn, F., Sidiropoulos, N., Faloutsos, C., Siegel, E., and Protopapas, Z.: *Fast nearest neighbor search in medical image databases*. Tech. rep., 1998.

[10] Fu, A. W.-c., Chan, P. M.-s., Cheung, Y.-L., and Moon, Y. S.: Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. *The VLDB Journal*, **9**(2), (2000), 154–173, URL `https://doi.org/10.1007/PL00010672`.

[11] Carter, J. L. and Wegman, M. N.: Universal classes of hash functions. *Journal of computer and system sciences*, **18**(2), (1979), 143–154.

[12] Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In *Vldb*, vol. 99, 1999, pp. 518–529.

[13] Rajaraman, A. and Ullman, J. D.: *Mining of massive datasets*. Cambridge University Press, 2011.

[14] Kovacs, L. and Szabó, G.: Automated learning of the morphological characteristics of the Hungarian language for inflection and morphological analysis.

[15] Sjögren, R.: VP-Tree. URL `https://github.com/RickardSjogren/vptree`. Undefined: undefined Copyright 2017 Rickard Sjögren Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

AN EXTENDED PRODUCTION PLANNING MODEL BASED ON THE SIMULATION OF HUMAN CAPABILITIES

Norbert Tóth
Bay Zoltán Nonprofit Ltd., Hungary
Division for SMART Systems
norbert.toth@bayzoltan.hu

Gyula Kulcsár
University of Miskolc, Hungary
Department of Information Engineering
iitkgy@uni-miskolc.hu

[Received . . . and accepted . . .]

Abstract. Digital mapping and modelling of the real systems in digitization solutions of Industry 4.0 are increasingly contributing to rising of the efficiency of production processes. The complexity of production and logistics systems is further increased by human resources, whose production capacity greatly influences the production process. This paper presents a flexible manufacturing system where employee skills affect system productivity. The values of performance indicators can be changed in a favourable direction using a discrete event-driven simulation model. This model uses an integrated, self-developed module that supports production scheduling and takes the production capabilities of workers into account for better use of the resources in the production system.

Keywords: simulation, production scheduling, human capabilities, digital model

1. Introduction

The common feature of the production companies – both multinational companies and SMEs – is that management aims to serve the customer needs to the highest possible standard, while fulfilling orders with short lead times and ensuring appropriate quality conditions [1]. The large number of orders results in a drastic increase in the number of finished product types. This has an impact both on the design of production systems and the operation of production and logistics processes. On one hand large-scale production is replaced by small and medium-sized series, on the other hand systems supporting production for individual needs are becoming more and more important, where the "one-piece" material flow will be dominating [2]. Meeting these needs can only be achieved with a high degree of flexibility and coordinated operation of the production and the logistics system. Opportunities offered by the Industry 4.0 solutions highly support this through the extensive cooperation of information and communication technologies (ICT), digitization and virtualization technologies [3]. Digital mapping of production and logistics processes of real production systems and the corresponding simulation based studies support the detection of bottlenecks [4]. Mapping of real processes is realized in the digital model at an abstraction level where evaluation of indicators characterizing the operation of the system and examination of influencing parameter values becomes possible [5]. Experiments by changing input parameter values can be performed in digital models, while leaving the smooth operation of real processes intact.

The importance of using simulation studies is increasing: a general model designed to solve a real problem contains several parameters which can be used to examine different operational strategies or to redesign system processes. The examined systems typically consist of discrete production and related logistics processes. Special softwares are available for modelling of these systems digitally. One of these is Tecnomatix Plant Simulation from Siemens, Plant Simulation is a general discrete-event-driven simulation development environment. It supports modelling and simulation of production, manufacturing and logistics processes with a wide set of objects [6], [7]. Using the SimTalk programming language, own algorithms, control procedures and functions may be created to support more realistic, detailed mapping of processes [8].

A common feature of most discrete manufacturing processes is that the operations must be performed on certain machines or workstations in a predefined order on the workpieces. One typical basic manufacturing scheme is the so-called "flow shop" (one-way) model, in which the same operations must be performed on the workpiece sets (jobs) in the same order on the same resources. There may be additional subcategories to this like the passing and no-passing versions. In the no-passing version, the execution sequences of jobs for machines can be different. In the passing version, only one execution sequence is given for all machines [9], [10], [11]. Another typical scheme is the "job shop" model, where each job can have a unique sequence of operations to be performed [12]. Both models may be flexible. If a group of machines is able to carry out an operation, then a machine-assigning process is also needed beside the job-sequencing task [13], [14], [15]. Even with a small number of machines, creating an optimal schedule with a polynomial runtime is not possible due to the complexity of the problem: the vast majority of models fall into the NP-hard problem class. In such cases, it is advisable to use fast heuristic, metaheuristic and search algorithms that give near-optimal solutions in a short runtime [16]. Development of complex models is necessary to support the production scheduling of real production systems, where the parameters, influencing factors and stochastic effects related to the logistics processes are also considered in addition to the parameters occurring in the production process.

Our research work summarized in this paper was induced by a real problem of a flexible assembly system. The following sections present a simulation model and a new heuristic scheduling algorithm for preparing daily production plans with the highest possible performance. The specialty of the system is that the model includes the individual assembly capabilities of human resources (persons). This extension further increases the complexity of the problem. The efficiency of our extended decision-making algorithm is verified by simulation running results.

2. Problem description

2.1. Operations and workplaces

Nowadays, increasing the competitiveness of companies plays a top priority in fulfilling customer orders in time. Besides, maximizing the performance of production processes, minimizing non-productive logistics processes, and reducing losses are also important. One of the losses in the production process is the changeover time related to switching between product types. Production scheduling can reduce the number and time of changeover and setup activities if the schedule is near to the optimal product sequence. As a result of changing customer needs, flexible production structures come to the fore, where production process involves the manufacturing of different product types at the same time. In additional, the unique process plan is given for each product type and they can differ in number or in type of operations. Each operation is linked to a machine or a group of equivalent machines where the necessary operations can be performed.

As a result of the application of Industry 4.0 technologies, individual devices can operate autonomously and communicate to each other on the network. As both the machines and the workpieces are identified, each machine can send messages to the next machine in the process flow to get prepared for the workpiece and reduce changeover time. Due to the different product types moves in the system at the same time, complex material flow relations are created. Fig. 1 illustrates the "job shop" problem in the form of a directed graph for 4 products and 5 machines as an example. Table 1 shows the unique sequence of operations for the product types.

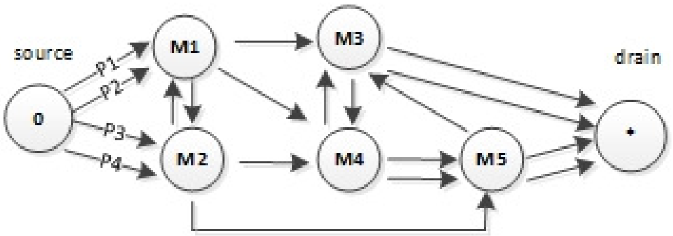


Figure 1. Job shop production model

Table 1. Order of operations by product type

	M1	**M2**	**M3**	**M4**	**M5**
P1	O_2,t_1	O_2,t_2		O_3,t_3	O_4,t_4
P2	O_1,t_5	O_3,t_6		O_2,t_7	
P3	O_2,t_7	O_1,t_8	O_3,t_8		
P4		O_1,t_8	O_3,t_9	O_4,t_{10}	$O_2,t_{11};O_4,t_{12}$

Each operation (O_i) has a processing time (t_j) that is required for a given operation of a given product type on a given machine.

2.2. Human resources

Several material flow relations exist for each machine as indicated on Fig. 1 Instead of using costly automated material handling, this task is performed by human labor: skilled workers select the next workpiece from the input buffer (source) and move the workpiece between the individual assembly stations (machines) according to the specified order of operations of the product type. The operations are performed at the stations. One worker at a time can perform one operation on a suitable assembly station; thus, if an assembly station is busy, the worker waits until it is released. After finishing the last operation, the assembled product is placed in the finished product container (drain). The cycle then starts again.

Consequently, the workers implement the "one-piece" material flow in the investigated system. The assembly process on workstations cannot be interrupted: workers take breaks only after the product has been placed in the finished product container (at the end of the running cycle). However, the many advantages of using human resources, the negative effects also appear in the production processes: the fulfilment of shift-level / daily / weekly production plans depends on the stochastic effects in the system, which stem mostly from the uniqueness of people. Each operation has a predefined norm time that serves as the basis for the preparation of production plans. From these plans shift-level production plans are generated, typically evenly distributed over the shifts for a given period. Although workers are expected to adhere to the standard time, some differences can be discovered due to the worker's manufacturing and assembly skills. While a rookie is usually not able to adhere to the norm times, more experienced ones perform the operations within the norm time. There may also be differences between those working night and day shifts. Thus, working skills can be interpreted as a percentage of the norm time (average time) of performing a given operation. The complexity of the production process further increased with the involvement of working skills, which also affects production planning, scheduling models and algorithms. This complexity justifies the use of simulation models and methods, with which the indicators of the productivity and efficiency can be evaluated depending on the different production plans and employee skills.

3. Digital model of the production system

3.1. Model objects

A new digital model had been developed before the examination of the presented production system. Real production and logistics processes are mapped in the model as follows:

- Modelling of production equipment (workstations, machines): There are 50 production equipment (A1-A50) with a fixed location in the production system. The cooperation of the production equipment and the worker results in the elementary operation of the manufacturing process on the workpiece, where an essential property of the workpiece changes. A controlled series of property changes make up the manufacturing process, which transforms the workpiece from the initial state to the finished state. The most important parameter of the operation is the processing time depending on the skills of the worker, which can be derived from the standard time for the execution of a specified operation of a fixed product on a given workstation (machine) (1).

$$t = t_{p_{ik}}^{A_j} + t_{p_{ik}}^{A_j}(1 - B_{p_{ik}}^{Operator_l}/100) \tag{3.1}$$

  where:
  – A$_j$: a j machine;
  – p$_{ik}$: operation k of product type i;

  - $t_{p_{ik}}^{A_j}$: standard time of operation k of product type i on machine A_j;
  - Operator$_l$: operator l;
  - B: production skill of operator;
  - $B_{p_{ik}}^{Operator_l}$: describes skills in percent of standard time of operation k of product type:
    - If the operator can maintain 100% of the norm time for the given operation, then the operation time of the operation is the same as the norm time.
    - If the operator is less than 100% able to meet the norm time of the given operation, the operation time increases as the operator works slower, so the time of the operation will be longer than the norm time.
    - If the operator can carry out the operation shorter than the norm time of the given operation (more than 100%), the operation time of will be shorter than the norm time.
- Employee modelling: Workers travel between workstations (machines) on a predefined route at a specified average speed. The worker selects the workpiece on the input storage object (”Start”) and then visits the workstations one by one according to the sequence of operations associated with the product type. After the last operation, the finished product is placed on the finished product storage object (”End”). Employees have breaks of predetermined lengths, which they use on decision. Breaks can used only after the finished product has been delivered. As multiple operators work at the same time, queues can form in case of shared access workstations. The operator remains at the station until the current operation is performed. The processing time depends on the actual operator capabilities. Operations are non-interruptible processes. There are equivalent machines that form a group with the same properties (certain operations can be performed on all equivalent machines). In the model, 23 operators (Operator1, ..., Operator23) were implemented for the simulated production environment, with different (very, medium, less efficient) capabilities.
- Product modelling: There are 15 product families and 100 product types in the model. Operation sequences are predefined for product families. The number and order of operations performed on the product types are the same as on product families as types are specialized representation of families. However, norm time of operations may be individually defined.

- ”Start” object (source): Functions as an input container, it stores the workpieces awaiting production, from which the operator chooses one.
- ”End” object (drain): An object that implements the storage of finished products. The finished product is placed here by the operators.

The Plant Simulation model of the production system is shown in Fig. 2 along with the production equipment, workers, and product types currently manufactured. Workers travel on a dedicated route. Paths appear where there is material flow between two workstations. Sankey diagram for two product types is also shown as an example. The thickness of the lines illustrates the size of the volume produced from the product types and indicates the workstations belonging to the technological process of the product type. Large amount of basic data is required for the operation of the model and the examination of the processes in the system, which is stored in a structured form in an external database. In addition to the properties of the basic objects of the model (products, machines, workers), the basic data also include the material flow relations and logical relationships between them. During the implementation of the model, following the previously developed concept model [4], quick adaption of the structure by modifying the basic data is possible. The structure of the model changes automatically, according to the modified basic data. Workstations, workers, products are created or deleted, the properties and data members of the model objects are dynamically updated.

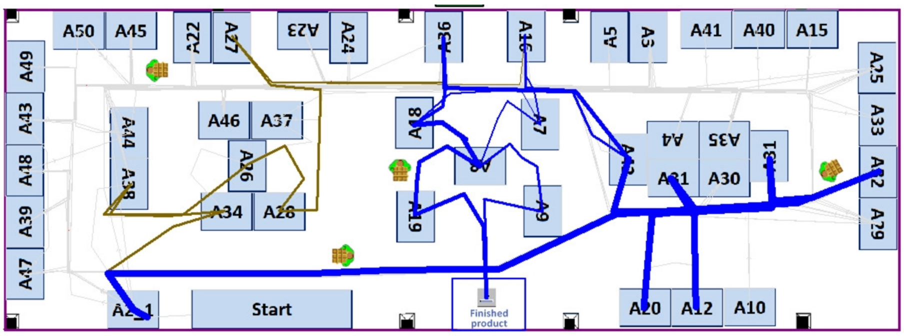


Figure 2. Simulation model

The following consistent data and data structures are required for the operation of the outlined dynamic behaviour digital model:

- Lists of employees are assigned to the shifts. This data structure represents an individual shift schedule of employees.
- List of product families, product types.

- Shift schedule: Cyclic change of 8-hour or 12-hour shifts in daily frequency. In a given shift, a given team of workers with different abilities is available.
- Sequence of production operations of product families, assigning operations to workstations where the given operation can be performed.
- Standard (norm) time for product type operations, which represents the expected operation time.
- Matrix of standard time per operation of product types and capabilities of operators: an element of the matrix shows the percentage of the given operator's ability to comply with the operation time (norm time) of the specific operation.
- Average capability matrix of operators for each product family: an element of the matrix specifies the percentage by which an operator can keep norm time of any operation belonging to a given product family.
- List of production plans:
  - Number of products per product type to be produced during the period.
  - Number of products per product type in shifts to be produced during the period.

The efficiency of the production process and the fulfilment of the production plan are greatly influenced by the order and number of different types of shifts in the examined period, as well as the workers and their production capabilities assigned to shifts.

3.2. Simulation study of the production system

The purpose of our simulation study is to analyse the efficiency and performance of the production system under changing conditions. The first step in the test is to set the current value of input parameters of the model. Currently, two main input parameter sets can be specified:

- The period under review, which defines the order and number of shifts, the operators working in each shift type and their capabilities.
- Production plan for the period considered. This is possible in two ways:
  - The quantity to be produced refers to the entire period under investigation, from which the production plans for each shift are automatically generated by distributing the products evenly depending on the number of shifts.
  - The quantity to be produced can also be specified directly per shift.

Based on the simulation results, statements are made on the performance of the production system, the utilization of the employees, and the number of products produced. One of the most important aspects is the fulfilment of the production plan, therefore one of the main indicators is the number of products produced. The other indicator is the free capacity of the production system over time. The following conclusions can be drawn from the relationship between these two indicators:

1. If the production plan is not fulfilled (there are unmanufactured pieces) and there is significant free time left, then not all batches were manufactured in some shifts due to the workers production capabilities or inadequate management decisions.
2. If the production plan is not fulfilled (there are unmanufactured pieces) and there is no significant free time left, then the production plan is over-planned, i.e., the planned quantity cannot be produced in a given period.
3. If the production plan is fulfilled and there is significant free time left, then the production is under-planned; there are free worker capacities and further pieces can be produced.
4. If the production plan is fulfilled and there is no significant free time left, the production plan adequately loads the system resources and the distribution of workloads is close to the optimum.

However, not only the distribution of the workloads by shift influences the performance of the system and the number of products produced. The management and decision-making strategy that determines the operation of the employees also plays a crucial role. Two events have been implemented in the model that affect employee activity and production system performance:

1. Assigning the product types to be manufactured and the operators to the "Start" object, where the product types to be manufactured in a shift form a queue. The operator selects one of the waiting workpieces in the queue. As there are product families that he is unable to produce at all due to his skills, the operator's default selection strategy is to select the first workpiece from the queue that he can produce. Despite this is a very simple binding, it can have several effects on the production process. The production system is highly sensitive for the order of the production queue. If the jobs behind each other belong to the same product family and the workers can produce them, then the workers follow the same technological path, visiting the same workstations in a row. This may result worker queues at workstations, increasing turnaround time and reducing efficiency.
2. Choice between equivalent machines: in certain cases (such as finishing operations), more than one machine is usable to perform the same type of operation. In such cases, the workers decision strategy is based on a

penalty function, calculated for the available machines. This function also considers the number of products produced.

Simulation studies proved that productivity of the examined production system and adequate utilisation of resources depend on the sequence of jobs in the queue due to the binding of the workers and the workpieces to be produced, in addition to the many influencing parameters and decision strategies.

3.3. Simulation results

The simulation model is suitable for examining the fulfilment of the production plan. The study covers 7 days with 14 shifts, where one shift is 12 hours length. The assignment of workers to shifts and their abilities were predefined. The input parameters of the model are as follows:

- Production plan for the period is indicated in Table 2:

Table 2. Production plan

Product type	**Quantity [pcs]**
_1607	910
_0601	140
_159A	238
_1626	420
_06A8	126
_15A7	70
_1257	714

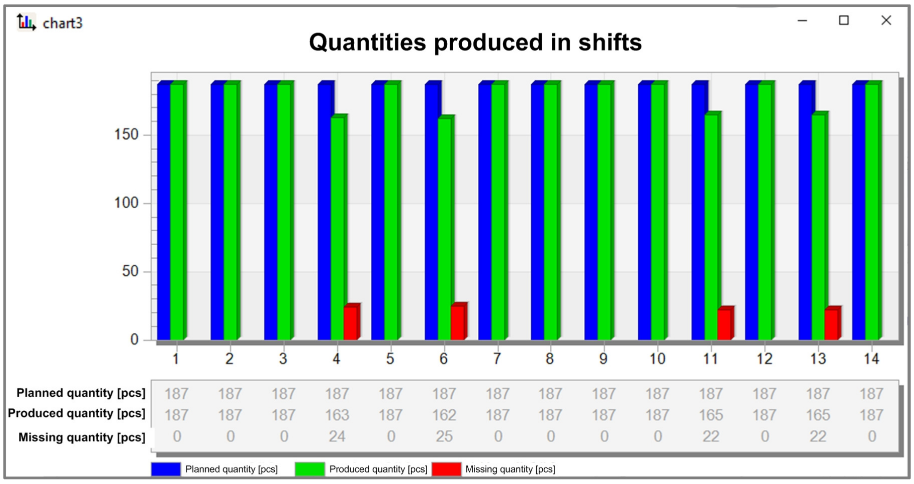
- Time period of the study, sequence of shifts. The production plan is evenly distributed among the shifts based on the number of shifts (with rounding).
- Type and usage of employee skills:
  - using theoretical operator skills: disregarding the production capabilities of the operators, each operator can produce each product type within 100% of standard time.
  - using average operator skills: the operating time can be calculated from the norm time and the average ability of the worker.
- KPIs in the system:
  - number of products produced
  - remaining free production capacity in time (remaining time is the sum of the remaining times at the end of the shifts when new products are not produced).

Results for the two investigated scenarios (theoretical and average operator skills) are summarized in Table 3.

Table 3. Comparison of simulation run results

	Theoretical skills	**Average skills**
Planned quantity [pcs]	2618	2618
Produced quantity[pcs]	2562	2525
Remaining time [min]	835	339

Table 3 indicates that the produced quantity is less than the planned quantity even if the workers perform with the maximum theoretical production performance. If the simulation runs with average capabilities, the amount produced is decreased because workers are unable to perform operations during norm time. Although this increases the turnaround time of the products, the system has time reserves in both cases. In some cases, the production plan for a shift was fulfilled before the end of the shift. Operators are waiting until the end of the shift in such cases. The time reserve of a shift is defined as the amount of time remaining (free production capacity in time): the difference in the useable time of the shift and the time required to complete the production plan. Analysing quantities produced per shifts (Fig. 3) and free capacities in time (Fig. 4) show that some shifts are overloaded (4, 6, 11, 13), its workers are unable to meet the requirements in the production plan.



Planned quantity [pcs]	187	187	187	187	187	187	187	187	187	187	187	187	187	187
Produced quantity [pcs]	187	187	187	163	187	162	187	187	187	187	165	187	165	187
Missing quantity [pcs]	0	0	0	24	0	25	0	0	0	0	22	0	22	0

Figure 3. Quantities produced in shifts

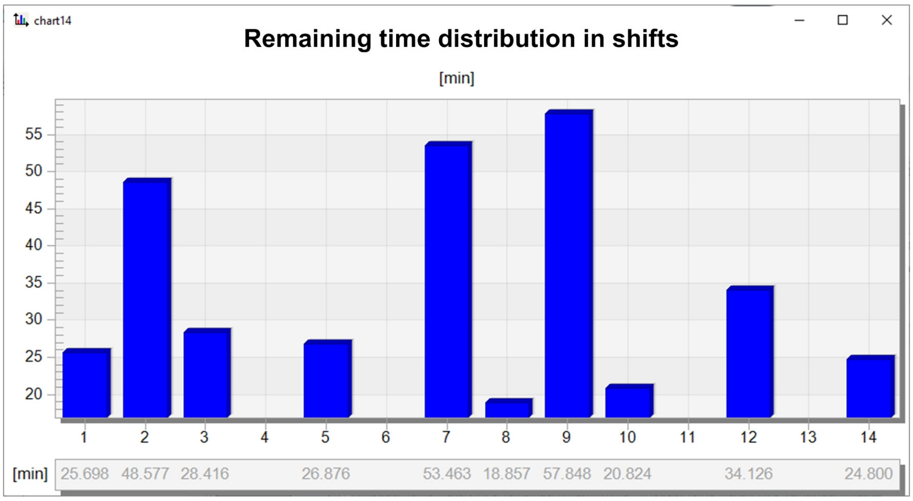


Figure 4. Remaining time distribution in shifts

4. The new heuristic method of the developed Production Plan Optimization (PPO) module

Based on Fig. 3 and Fig. 4 a production plan optimization (PPO) module was developed and implemented in Plant Simulation environment to increase the number of products produced. This PPO module creates an initial production plan based on an even distribution of the products to be produced in shifts. After simulation of the initial solution, an iterative re-planning phase starts. A new heuristic load-balancing method assigns the unmanufactured products to suitable shifts, in which there is available free production capacity (remaining time) and the active employees can perform the additional loads.

The aim of PPO is to achieve an improved production plan that results excellent performance indicators by modifying the evenly distributed loads of shifts and taking employee skills into account. The highly simplified algorithm of the PPO module is shown in Fig. 5 In the preparatory (iteration 0) step, the module evenly distributes the specified production plan among all shifts. Then it runs the simulation and evaluates the performance indicators. If there are unmanufactured products, these are assigned to the first suitable shift in which there is free production capacity (remaining time) and workers can produce these product types.

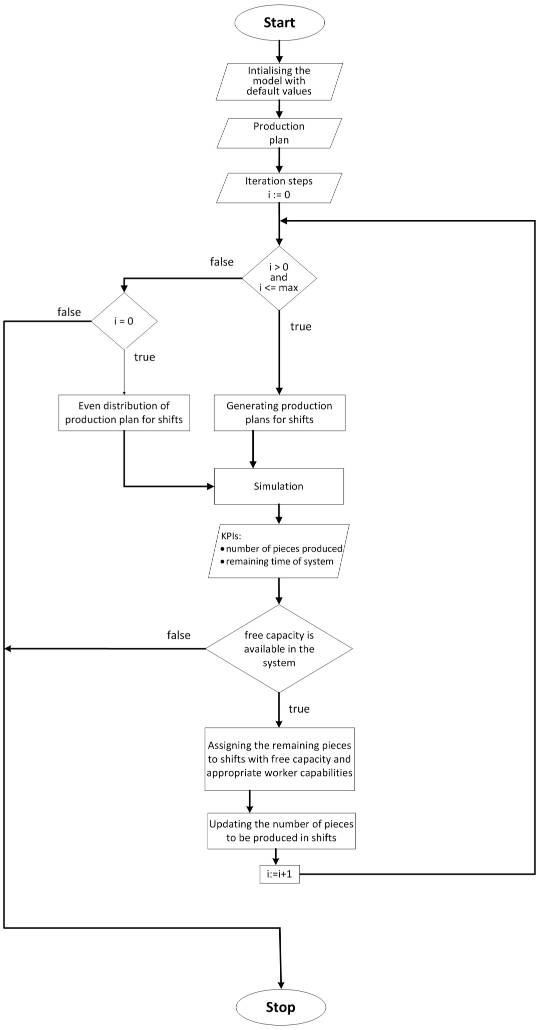
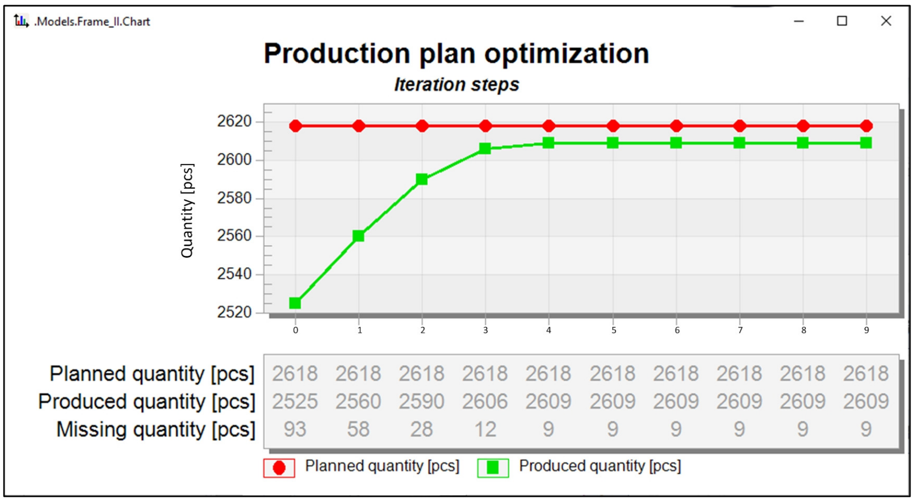

Start
Intialising the model with default values
Production plan
Iteration steps
i := 0
false
i > 0 and i <= max
true
false
i = 0
true
Even distribution of production plan for shifts
Generating production plans for shifts
Simulation
KPIs:
• number of pieces produced
• remaining time of system
false
free capacity is available in the system
true
Assigning the remaining pieces to shifts with free capacity and appropriate worker capabilities
Updating the number of pieces to be produced in shifts
i:=i+1
Stop

Figure 5. Flowchart of the iterative search algorithm of PPO module

The simulation is run again with the modified production plans, and the performance indicators are evaluated. This iterative process is repeated if there are unmanufactured products and there is free production capacity (remaining time) in the production system, or the number of iterations reached the maximum value. The algorithm can improve the system's performance even with a small number of iterations (Fig. 6). The number of pieces produced increased from 2,525 to 2,609, which indicates the better use of available time base of the production system. This is also shown by the decrease in the free time capacity of the system: from 393 minutes to 85 minutes.



Planned quantity [pcs]	2618	2618	2618	2618	2618	2618	2618	2618	2618	2618
Produced quantity [pcs]	2525	2560	2590	2606	2609	2609	2609	2609	2609	2609
Missing quantity [pcs]	93	58	28	12	9	9	9	9	9	9

Figure 6. PPO module iteration steps

The result of the iterative improving (re-planning) algorithm is a detailed production plan that specifies the numbers of pieces of the product types to be produced in each shift. As an illustrative example for the investigated case study, the detailed production plan is indicated in Table 4.

Table 4. Number of pieces produced in shifts

Shift types	**1.A**	**2.B**	**3.D**	**4.C**	**5.D**	**6.C**	**7.B**	**8.A**	**9.B**	**10.A**	**11.C**	**12.D**	**13.C**	**14.D**
_1607	69	68	67	56	68	56	70	68	70	66	57	68	56	71
_0601	10	11	11	9	10	9	10	10	11	10	9	11	9	10
_159A	18	19	17	15	18	15	18	17	18	18	15	17	15	18
_1626	30	33	32	26	31	27	32	30	32	31	27	30	27	32
_06A8	9	10	10	8	9	8	10	9	9	9	8	10	8	9
_15A7	6	5	5	4	5	4	6	6	5	5	4	5	4	6
_1257	53	54	53	44	54	43	55	54	55	52	43	54	44	56
Even distribution	187	187	187	187	187	187	187	187	187	187	187	187	187	187
PPO module	195	200	195	162	195	162	201	194	200	191	163	195	163	202

The implemented heuristic algorithm reduced the number of pieces to be produced in some shifts, while it was increased in other shifts. Type C shift workers appeared in the system as bottlenecks. Their low production capability worsened the productivity of the shift.

5. Summary and further development

In this paper, a special model of a flexible production system was presented, where human resources, production efficiency and employee skills play key roles to achieve the maximum productivity that can be expected from the production system. This direction of modelling and development is necessary because the human factor also has a strong influence on the performance of the flexible production process. In conclusion, the human factor must be integrated into the models and methods of organization, planning and scheduling of production to achieve the desired effect of the planned changes that aimed to increase the efficiency.

For this purpose, a digital simulation model was developed, which is suitable for considering the impact of the human factor at the levels of production planning, scheduling and control. A new heuristic algorithm was used, which increases the performance of the production system by improving the production plan for each shift. Even better results may be reached with optimizing the production sequence by using a novel strategy to assign dynamically the workpieces to the workers. This strategy can be put into practice if the control system supports each worker to select the workpiece that best suits his production ability and takes the product types already in production into account to reduce waiting times at workstations. This concept can be ensured by maximizing the heterogeneity of the product types produced at same time. It is recommended that the worker should choose a product type that few workers can produce. It is preferred that the control system also considers the production capability of all the other workers in order to effectively manage the dynamic manufacturing processes.

This proposed approach ensures the continuous work of the workers and increases the number of pieces produced.

In the future, several additional considerations must be taken in order to determine automatically quasi-optimal shift-level production sequences. We are developing a multi-objective priority-based rescheduling algorithm to extend the presented simulation model and solutions.

REFERENCES

[1] Szakál, F., Józsa, L.: *A 21. század fogyasztója, avagy mi a fontos a fogyasztónak a modern világban*, ”Kulturális gazdaság” Kautz Gyula Emlékkonferencia elektronikus formában megjelenő kötete, Széchenyi István Egyetem, Győr, 2019, ISBN 978-615-5837-34-0.

[2] Tóth, N., Ladányi, R., Garamvölgyi, E.: *Elaborating Industry 4.0 compatible DSS for enhancing production system effectiveness*, IOP Conf. Ser.: Mater.

Sci. Eng., Vol. 448 012040, Kecskemét, Hungary, 2018, https://10.1088/1757-899X/448/1/012040.

[3] Russmann, M., Lorenz, P., Gerbert, P., Waldner, M., Jastuss, J., Hengel, P., Harnisch, M.: *Industry 4.0: the future of productivity and growth in manufacturing industries*, The Boston Consultin Group report, 2015.

[4] Tóth, N.: *Termelési folyamatok intenzifikálását célzó új módszer bemutatása az Ipar 4.0 lehetőségei alapján*, Műszaki Tudomány az Észak-kelet Magyarországi Régióban, Konferencia Kiadvány, Debrecen, 2019, pp. 396-399., ISBN978-963-7064-38-8.

[5] VDI-Richtlinie 3633 Blatt 1.: *Simulation von Logistik-, Materialfluss- und Produktionssystemen-Grundlagen*, Düsseldorf, VDI-Verlag, 1993.

[6] Siderska, J.: *Application of Tecnomatix Plant Simulation for modeling production and logistics processes.* Business, Management and Education, 14(1), 2016, pp. 64-73., https://doi.org/10.3846/bme.2016.316.

[7] Tamás, P., Illés, B., Tollár, S.: *Simulation Of A Flexible Manufacturing System*, Advanced Logistic systems, University of Miskolc, Department of Material Handling and Logistics, Vol. 6(1), December, 2012, pp. 25-32., HU ISSN 1789-2198.

[8] SIEMENS AG.: *Tecnomatix Plant Simulation Help*, 2017.

[9] Kulcsár, Gy., Erdélyi, F.: *A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks*, International Journal of Computational Intelligence Research, 3 (4), 2007, pp. 343-351., DOI: 10.5019/j.ijcir.2007.115.

[10] Kulcsár, Gy.: *Ütemezési modell és heurisztikus módszerek az igény szerinti tömeg-gyártás finomprogramozásának támogatására*, Doktori (PhD) értekezés, Miskolci Egyetem, Miskolc-Egyetemváros, 2007.

[11] Kulcsárné, F. M.: *Kiterjesztett modellek és módszerek erőforrás-korlátos termelésütemezési feladatok megoldására*, Doktori (PhD) értekezés, Miskolci Egyetem, Miskolc-Egyetemváros, 2017.

[12] Kulcsár, Gy., Kulcsárné, F. M.: *Kiterjesztett termelésprogramozási modell erőforrás-korlátos ütemezési feladatok megoldására*, Multidiszciplináris tudományok, 4. 1. sz., 2014, pp. 19-30.

[13] Botta-Genoulaz, V.: *Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness.* International Journal of Production Economics, Vol. 64, 2000., pp. 101–111., https://doi.org/10.1016/S0925-5273(99)00048-1.

[14] Low, C.: *Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines*, Com-puters and Operations Research, Vol. 32, 2005., pp. 2013–2025., https://doi.org/10.1016/j.cor.2004.01.003.

[15] Demir, Y., İşleyen, S. K.:*Evaluation of mathematical models for flexible job-shop scheduling problems*, Applied Mathematical Modelling, Volume 37, Issue 3, 2013, pp. 977-988., ISSN 0307-904X, https://doi.org/10.1016/j.apm.2012.03.020.

[16] Fogarasi, G., Tüű-Szabó, B., Földesi, P.: *Az Utazó Ügynök Problémára alkalmazható diszkrét memetikus evolúciós metaheurisztikák összehasonlítása*, Logisztika-Informatika-Menedzsment Vol. 4, number 1, 2019., pp. 15-30., doi: 10.29177/LIM.2019.1.15.

Production Systems and Information Engineering
Volume 9 (2020), pp. 36–53

STUDENT ACADEMIC PERFORMANCE PREDICTION

Jawad Alshboul
University of Miskolc, Hungary
Department of Information Technology
alshboul.jawad@student.uni-miskolc.hu

Erika Baksa-Varga
University of Miskolc, Hungary
Department of Information Technology
vargae@iit.uni-miskolc.hu

[Received . . . and accepted . . .]

Abstract. Given the increasing number of students who attend traditional and non-traditional classes that deploy internet-based educational resources and environments, large volumes of data are being generated on a daily basis. As a result, more researchers are now working with Educational Data Mining (EDM) methods to understand learning processes and behaviors of learners. The problem that led to this research is the need to make use of unused data that is collected during education and learning processes by gaining insights in order to support students in regards to their academic performance and in taking actions to prevent or warn students from failure. The main focus of this research is on how EDM can support student learning in regards to student academic performance, engagement, and intervention. The research mainly addresses the appropriate EDM methods used to predict student academic performance. Modeling and evaluation of several classifiers were conducted. As a result, Random Forest classifier has been chosen as the best model to be deployed in an interactive R Shiny application.

Keywords: educational data mining, academic performance prediction, classification

1. Introduction

Due to the growing use of educational resources and technologies, educational data are being generated in huge amounts on a daily basis. Data-driven decision making (DDDM) refers to the systematic processes of collection, analysis, and interpretation of data to help in decision making [1]. Educational data can be used in DDDM at different educational levels to achieve effective educational data decision making. DDDM for educational data can be related to educational resources, and human resources decisions.

Data-Driven Education enables institutions to leverage educational data to get insights about teaching-learning process and to make data-driven educational decisions based on student needs [2]. DDDM includes exploiting available data, such as the kind offered in virtual learning environments or Learning Management Systems (LMS), to make teaching decisions [3, 4]. Values underlying educational data mining are to analyze student learning data and its contexts in order to better understand and personalize student learning experiences [5, 6].

According to [7], Data Mining (DM) is a computerized information system dedicated to handle large amounts of data, produce information, and discover hidden patterns. The demand on using DM in educational settings led to the establishment of Educational Data Mining (EDM) as a new field of knowledge and line of research [8]. The growing acceptance of the emergent field, EDM, is due to its ability to elicit valuable insights from data for either students or staff [9]. The authors in [10] define EDM as a multidisciplinary field of study that combines skills and knowledge from machine learning, statistics, DM, psychology, information retrieval, cognitive science, and recommender systems techniques to support resolving issues related to education.

The main reason for the late emergence of data mining in education, compared to all other fields, was that the availability of large educational datasets in machine readable formats emerged later in education [11].

2. Educational Data Mining Methods and Applications

2.1. Regression Techniques

There is a number of regression algorithms such as: Single Linear Regression, and Multiple Linear Regression. The regression technique is used to predict values and it has been applied in education domain to: predict students' grades [12], and predict academic GPA of graduated student [13].

2.2. Classification Techniques

There is a number of classification algorithms such as: Decision Trees, Neural Networks, Logistic Regression, and Nave Bayes classifiers. Classification techniques have been applied in education domain to: analyze the academic performance of undergraduate students [14], assess how effective EDM techniques are for students early prediction failure [15], and develop a model to prevent academic dropout [16].

2.3. Clustering Techniques

There are different clustering algorithms such as: K-Means and Expectation Maximization. Clustering techniques have been applied in education domain to: generate a model for student dropout by exploring student categories and characteristics [17], group university students into careers by analyzing their performance and outcomes of the self-evaluation test beginning from their first year [18], associate students and teachers [19], and group competent students of an educational institution in regards to their skills and abilities [20].

2.4. Association Rules Techniques

The association rule mining techniques are applied to identify associations or dependencies between attributes in the datasets [21]. Association Rules have been applied in education domain to: propose a quantifiable measure that shows degradation in regard to students expected performance [22], analyze students' performance based on real time patterns in students' data [23], investigate on association between self-esteem and performance of students [24], and discover the impact of teaching on improving how student performs [9].

2.5. Social Network Analysis and Visualization Techniques

Social Network Analysis (SNA) and Visualization can reduce the size of the datasets and their complexity in case they are multidimensional datasets. SNA and visualization have been applied in education domain to: introduce a model based on visual analytics, and learning analytics, in addition to a tool, to perform confirmatory and exploratory data analysis through interaction between information gathered [25], process the interaction networks of students in a forum [26], and check the progress of online collaborative learning and provide informed interventions when needed [27].

2.6. Process Mining Techniques

Process Mining techniques are used to deal with log files and events and they have been applied in education domain to: analyze events flow logs in an adaptive learning model [28, 29], and provide feedback on the basis of behavioral data [30].

2.7. Text Mining Techniques

Text Mining techniques are used to deal with unstructured data by capturing key terms and uncover hidden patterns. Text Mining techniques have been applied in education domain to: analyze students' online interaction via online questions and chat messages [31], extract knowledge from students' evaluation comments that help instructors and administrators obtain understanding of student sentiments and views [32], and rate educational institute faculty members based on the feedback submitted by students [33].

2.8. Outlier Detection Techniques

Outlier Detection is used to check whether there is any deviation in any observation away from all other observations using data mining algorithms based on association, classification, clustering, visualization, or statistics-based approach. It has been used in education domain to: discover any anomaly or abnormal observations [34, 35], and predict dropouts by clustering outlier data with unsupervised learning [36].

2.9. Student Academic Performance Prediction

Student academic performance prediction has been an important research topic for years since students and institutions can benefit from discovering patterns and insights hidden within learning data. Institutions can benefit from it by improving the effectiveness of academic facilities available to their students in order to increase the rates of students who are successful in completing their programs or courses of study. Furthermore, findings can be used to deliver solutions, suggestions, or advices to students to enhance how they perform in the future.

A review of literature on the methods used for student academic performance prediction was conducted. The search focused on Scopus, IEEE, Google Scholar, and ACM for years 2009 to 2019. The number of relevant articles used for the synthesis, after excluding the articles that did not describe the data sets attributes or methods used, is 157 articles.

Table 1 shows some statistics related to the modeling techniques that have been used in the studies related to performance prediction. It is shown that the mostly used classification modeling techniques are: Decision Trees, Bayesian-Based, Neural Networks, Support Vector Machines, Ensemble Methods, K-Nearest Neighbor, and Logistic Regression, respectively.

Table 1. Statistics of modeling techniques used

Modeling Technique	**Count**	**Percentage**
Decision Trees	79	50.3 %
Bayesian-based	65	41.4 %
Neural Networks	43	27.4 %
Support Vector Machines	34	21.7 %
Linear Regression	34	21.7 %
Ensemble Methods	29	18.5 %
K-Nearest Neighbor	24	15.3 %
Logistic Regression	22	14.0 %
Others (Hybrid, optimization, statistical, ..etc.)	12	7.6 %
Rule Induction	9	5.7 %

3. Research Methodology

3.1. Research Objectives and Research Questions

The research question (RQ) for each research objective (RO) is explained as follows:
RO1. To investigate the appropriate educational data mining methods used in predicting student academic performance.
RQ1. What are the appropriate techniques that are used to predict student academic performance?
RO2. To apply educational data mining methods to support academic intervention.
RQ2. How can educational data mining be used to support academic intervention?

3.2. Data Collection and Preparation

A quantitative dataset [37] has been chosen based on the literature review conducted to get answers for research questions. The significance of this dataset is due to adopting student behavioral features with academic data during the learning process.

The dataset used was collected from a multi-agent Learning Management System (LMS) called Kalboard 360 using Experience API (xAPI) web service [37].

An activity tracker tool called experience API (xAPI) was used to track learners. The dataset shown in Table 2 consists of 480 student records and 16

Table 2. Student academic performance dataset description

Category	Feature	Data Type	Description
Demographical Information	Nationality	Nominal	Student Nationality
	Gender	Nominal	Student Gender (female or male)
	Place of Birth	Nominal	Student Place of Birth (Jordan, Kuwait, Lebanon, Saudi Arabia, Iran, USA)
	Parent Responsible	Nominal	Student Parent (father or mum)
Academic Information	Educational Stages (School Levels)	Nominal	Stage student belongs such as (primary, middle and high school levels)
	Grade Levels	Nominal	Student Grade (G-01 → G-12)
	Section ID	Nominal	Student Classroom (A, B, C)
	Semester	Nominal	School Year Semester (First or Second)
	Topic	Nominal	Course Topic or Subject (Math, English, IT, Arabic, Science, Quran)
	Student Absence Days	Nominal	Student Days of Absence (Above-7, Under-7)
Parents Participation in Learning Process	Parent Answering Survey	Nominal	Parent Answering School Surveys or Not.
	Parent School Satisfaction	Nominal	Parent Satisfaction Degree about School (Good, Bad)
Behavioral Information	Discussion Groups	Numerical	Student Behavioral Interaction with E-Learning System.
	Visited Resources	Numerical	
	Raised Hand on Class	Numerical	
	Viewing Announcements	Numerical	
Target	Student Mark	Ordinal	L: Low-Level values from 0 to 69. M: Middle-Level values from 70 to 89. H: High-Level values from 90-100.

features in addition to the target variable which represents student academic performance. The 16 features are grouped into four categories: features that constitute Demographic Information, features that constitute Academic Information, features that constitute Parents Participation in Learning Process Information, and features that constitute Behavioral Information.

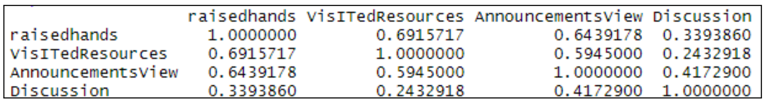
3.2.1. Data Preprocessing

Checking missing values, renaming some attributes, data type conversion of some attributes to factors, and correcting some misspelled country names by using the standard names were required to prepare data for the next steps.

3.2.2. Feature Selection

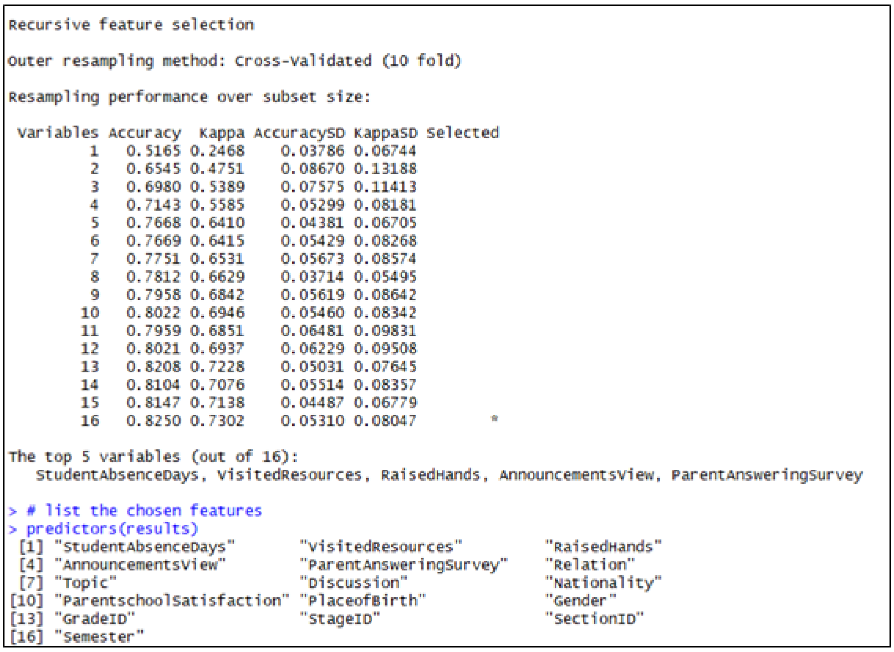
Feature Selection is the process used for selecting those dataset features that will contribute most to the prediction task. Correlation matrix and recursive selection are used for this purpose.

1. Correlation Matrix: Figure 1 shows the output for feature selection based on correlation matrix setting with a cutoff 0.75 for highly correlated attributes. It is clear that there are no two features correlated with at least 0.75 to be considered highly correlated.
2. Recursive Selection: Recursive Feature Elimination (RFE) method is used to identify the features that can be eliminated without affecting the accuracy of classification models. Figure 2 and Figure 3 show the result of performing REF on the features of the data set. Based on the result of performing REF on the features of the data set, it is possible to either keep or remove semester feature from the data set. Consequently, it is kept for the further phases in modeling. As a result, the 16 features will be used to build models for predicting student academic performance.



```
                  raisedhands VisITedResources AnnouncementsView Discussion
raisedhands         1.0000000        0.6915717         0.6439178  0.3393860
VisITedResources    0.6915717        1.0000000         0.5945000  0.2432918
AnnouncementsView   0.6439178        0.5945000         1.0000000  0.4172900
Discussion          0.3393860        0.2432918         0.4172900  1.0000000
```

Figure 1. Correlation matrix result



```
Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
         1   0.5165 0.2468    0.03786 0.06744         
         2   0.6545 0.4751    0.08670 0.13188         
         3   0.6980 0.5389    0.07575 0.11413         
         4   0.7143 0.5585    0.05299 0.08181         
         5   0.7668 0.6410    0.04381 0.06705         
         6   0.7669 0.6415    0.05429 0.08268         
         7   0.7751 0.6531    0.05673 0.08574         
         8   0.7812 0.6629    0.03714 0.05495         
         9   0.7958 0.6842    0.05619 0.08642         
        10   0.8022 0.6946    0.05460 0.08342         
        11   0.7959 0.6851    0.06481 0.09831         
        12   0.8021 0.6937    0.06229 0.09508         
        13   0.8208 0.7228    0.05031 0.07645         
        14   0.8104 0.7076    0.05514 0.08357         
        15   0.8147 0.7138    0.04487 0.06779         
        16   0.8250 0.7302    0.05310 0.08047        *

The top 5 variables (out of 16):
   StudentAbsenceDays, VisitedResources, RaisedHands, AnnouncementsView, ParentAnsweringSurvey

> # list the chosen features
> predictors(results)
 [1] "StudentAbsenceDays"       "VisitedResources"         "RaisedHands"
 [4] "AnnouncementsView"        "ParentAnsweringSurvey"    "Relation"
 [7] "Topic"                    "Discussion"               "Nationality"
[10] "ParentschoolSatisfaction" "PlaceofBirth"             "Gender"
[13] "GradeID"                  "StageID"                  "SectionID"
[16] "Semester"
```

Figure 2. Recursive feature elimination performed on the data set

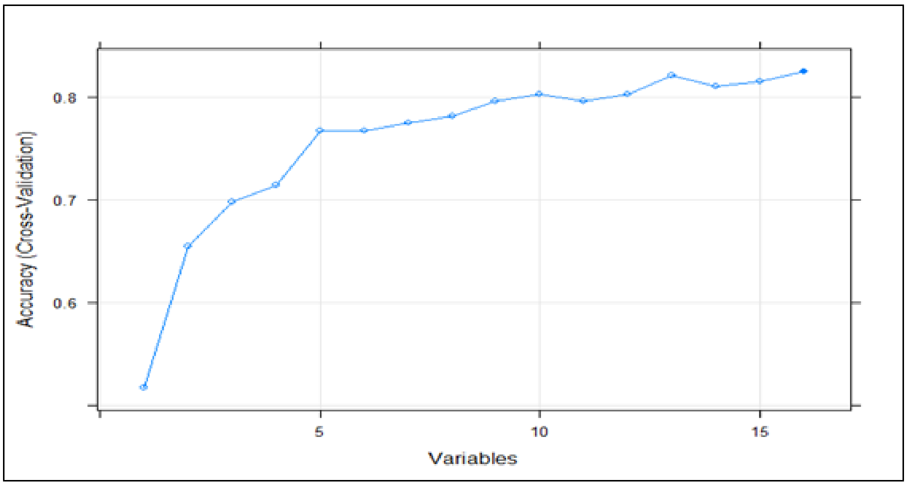


Figure 3. Accuracy based on importance of features in modeling classifiers

3.3. Modeling

The classification techniques used are: Decision Trees, Naive Bayes, K-Nearest Neighbors, Support Vector Machines, Logistic Regression, Ensemble Methods (Random Forest), and Neural Networks.

3.3.1. K-Nearest Neighbor (KNN)

It is a non-parametric, instance-based supervised learning algorithm, and easy to implement and understand but computationally expensive. KNN algorithm works as follows:

1. Choose K as the number of neighbors.
2. Select the K nearest neighbors of the unknown data point based on their Euclidean distances (Or other distance measure if the points are categories) from that unknown data point.
3. From the selected K neighbors, compute the number of data points in each category.
4. Allocate the new data to the category that has the largest count of neighbors among other categories.

3.3.2. Decision Tree (DT)

It is a supervised learning algorithm and It works for both continuous and categorical data. It splits the nodes based on all features then selects the appropriate split using some criteria such as Gini index, Information Gain, and Variance. DT algorithm works as follows:

1. Place all training examples at the root node.
2. Categorize the data set attributes.
3. Split examples based on specific selected attributes.
4. Select test attributes based on a specific measure like statistics or heuristic.
5. Stop when: all examples being members of the same class, no attributes left for partitioning, or no examples left for classification.

3.3.3. Support Vector Machines (SVM)

They are supervised learning algorithms that can be used for both classification and regression problems. SVM algorithm works in the following steps:

1. Plot each data point in an m-dimensional space where m represents the number of attributes in the dataset.
2. Perform classification by trying to find the suitable hyperplane that separates the classes.

3.3.4. Logistic Regression (LR)

It is a supervised learning algorithm that can be used for binary classification but can deal with multi-class classification problems as well by using one-vs-all principle. A logistic function called sigmoid function is used to map the outputs to probabilities. One-vs-all classification is performed by training M distinct binary classifiers in which each trained binary classifier can recognize a particular class. Consequently, those M classifiers are combined together to be used for multi-class classification.

3.3.5. Random Forest (RF)

It is an ensemble decision tree that creates and combines many decision trees. Creating and combining many decision trees allow weak decision trees on their own to be used in order to create a stronger decision tree with better accuracy. It is called random because the attributes are chosen randomly during model building and training. Furthermore, it is called forest because it takes outputs of many decision trees to create a better decision tree. RF algorithm works in the following steps:

1. Choose n samples randomly from the training set.
2. Grow a decision tree from the chosen sample by selecting a number of features randomly.

3. Split the node based on a chosen feature which has the highest information gain.
4. Repeat the previous steps K times (K represents the number of trees to be created).
5. Aggregate the trees and then choose the majority class based on voting.

3.3.6. Nave Bayes (NB)

It is a Bayes theorem-based supervised learning algorithm. It is called nave because it assumes that an attribute is independent in terms of probability to happen from all other features. NB formula is shown as follows:

$$P(C|f) = \frac{P(f|C)P(C)}{P(f)} = P(f_1|C) \times P(f_2|C) \times ... \times P(f_n|C) \times P(C) \quad (3.1)$$

Where C is the class and f_i is any feature.

3.3.7. Neural Networks (NN)

They are machine learning algorithms that are built based on the human brain. They can be used for multi-class classification problems by considering one-vs-all principle. An activation function is used to take a number of inputs to produce an output. Assume there are M classes, one-vs-all classification is performed by training M distinct binary classifiers in which each trained binary classifier can recognize a specific class. Consequently, those M classifiers work together to be used for multi-class classification.

3.4. Evaluation

A decision on the adoption of the EDM outcomes should be delivered based on the evaluation phase. The Confusion matrix will be used for calculating the correctness and accuracy of the model.

Since target variable classes are nearly balanced, accuracy will be used as a performance metric to compare the models. The Confusion matrix is an easy and intuitive metric used for finding the correctness and accuracy of the model. It is used for classification problems where the output would be of two or more types of classes. Some Terminology and derivations from a confusion matrix are shown as follow:

1. True Positive (TP): Cases that are TRUE and predicted correctly as TRUE.

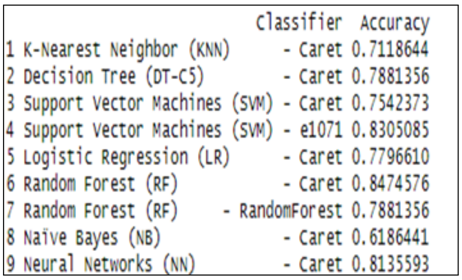
2. True Negative (TN): Cases that are FALSE and predicted correctly as FALSE.
3. False Positive (FP): Cases that are FALSE but predicted incorrectly as TRUE (known as ”Type I error”).
4. False Negative (FN): Cases that are TRUE but predicted incorrectly as FALSE (known as ”Type II error”)

Accuracy is a metric for evaluating classification models and it refers to the percentage of predictions that happen to be right. Based on the contents of the confusion matrix it is possible to extract the accuracy of the model as shown in following formula:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.2}$$

The research tools need to verify the reliability and validity. In quantitative research, enhancing and verifying of experiments are achieved through measurement of the validity and reliability [38]. To make sure that the study is reliable and valid, the experiments conducted using 10-fold cross validation repeated for 10 times.

Figure 4 shows the final result as a comparison based on the accuracy of each classifier. Random Forest (RF) implemented using the Caret package shows the best performance with 85% accuracy.



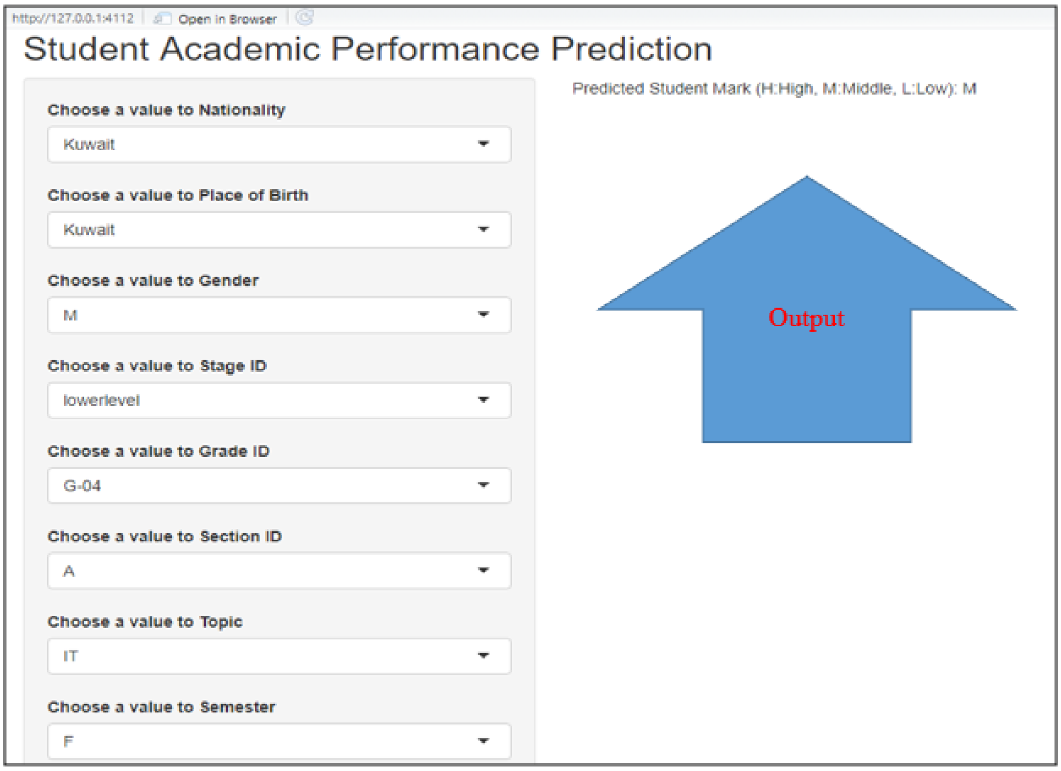
```
                          Classifier  Accuracy
1 K-Nearest Neighbor (KNN)      - Caret 0.7118644
2 Decision Tree (DT-C5)         - Caret 0.7881356
3 Support Vector Machines (SVM) - Caret 0.7542373
4 Support Vector Machines (SVM) - e1071 0.8305085
5 Logistic Regression (LR)      - Caret 0.7796610
6 Random Forest (RF)            - Caret 0.8474576
7 Random Forest (RF)     - RandomForest 0.7881356
8 Naïve Bayes (NB)              - Caret 0.6186441
9 Neural Networks (NN)          - Caret 0.8135593
```

Figure 4. Accuracy measure of each classifier

4. Discussion

4.1. Deployment

Deployment phase includes the outputs of the experiments as explained in Modeling and it shows a deployment of the best model, found during modeling



Student Academic Performance Prediction

Choose a value to Nationality: Kuwait

Choose a value to Place of Birth: Kuwait

Choose a value to Gender: M

Choose a value to Stage ID: lowerlevel

Choose a value to Grade ID: G-04

Choose a value to Section ID: A

Choose a value to Topic: IT

Choose a value to Semester: F

Predicted Student Mark (H:High, M:Middle, L:Low): M

Figure 5. Interactive web application using r shiny part 1

and evaluation phase, in interactive web application using R Shiny package. Figure 5 and Figure 6 show the user interface of the Shiny web application developed to deploy the predictive model.

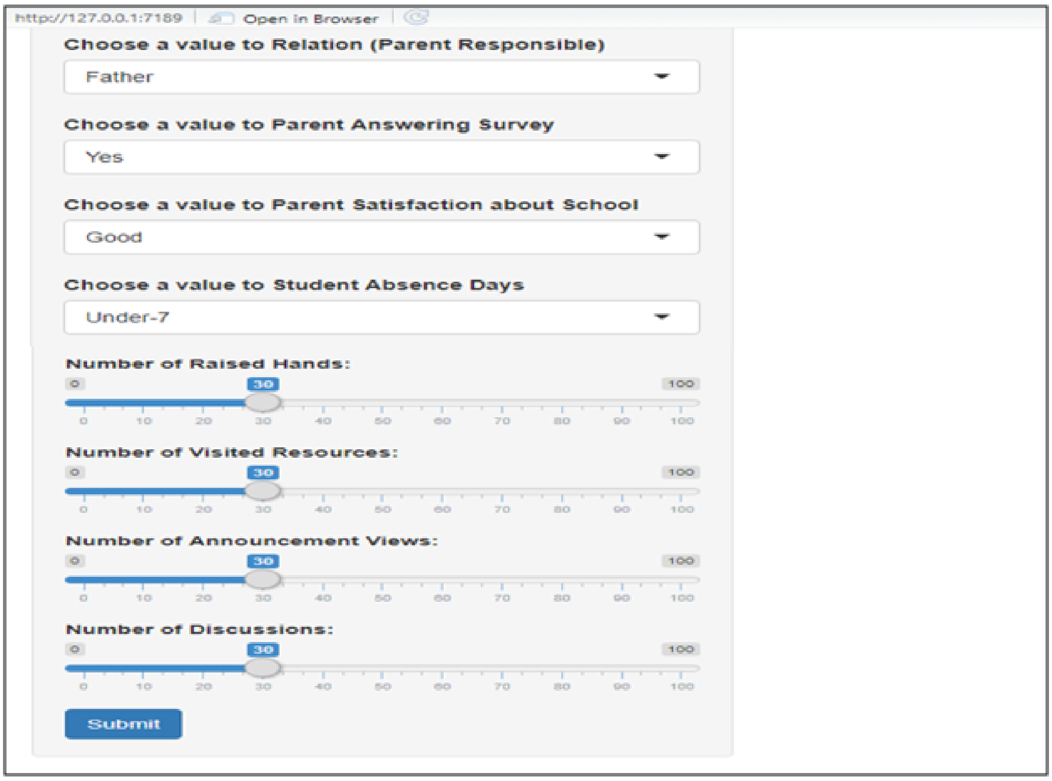
4.2. Findings

The final predictive model built using random forest revealed some interesting findings which are listed as follows:

1. Based on the literature review conducted, seven classifiers are chosen to model the dataset: Decision Trees, Nave Bayes, K-Nearest Neighbors, Support Vector Machines, Logistic Regression, Random Forest, and Neural Networks.
2. The best model in regards to accuracy is the one built with Random Forest using the Caret package.
3. The best model is deployed into an interactive R Shiny application.

4.3. Fulfilment of Research Objectives

RO1. Investigating the appropriate educational data mining methods used in predicting student academic performance has been achieved by describing the techniques of educational data mining first, then exploring educational data mining techniques used in predicting student academic performance in depth as discussed in Section 2. Educational Data Mining techniques explored are regression, classification, clustering, association rules, social network analysis and visualization, process mining, text mining, and outlier detection. A synthesis of literature related to the appropriate educational data mining techniques relevant to student academic performance prediction shows that there are two relevant educational data mining techniques which are regression (Linear Regression) and classification (Decision Trees, Bayesian-Based, Neural Networks, Support Vector Machines, Ensemble Methods, K-Nearest Neighbor, and Logistic Regression).



Choose a value to Relation (Parent Responsible)
Father

Choose a value to Parent Answering Survey
Yes

Choose a value to Parent Satisfaction about School
Good

Choose a value to Student Absence Days
Under-7

Number of Raised Hands: 30

Number of Visited Resources: 30

Number of Announcement Views: 30

Number of Discussions: 30

Figure 6. Interactive web application using r shiny part 2

RO2. To apply educational data mining methods to support academic intervention by applying the relevant educational data mining classification techniques explained in the synthesis of the literature review performed in Section 2 on an appropriate dataset and building a model using R Shiny to gain insights from learning data in order to support students in regards to their academic performance and in taking actions to prevent or warn students from failure which leads to grade improvement that in turn will drive the overall degree success as discussed in Sections 3.2, 3.3, 3.4, and 4.1.

5. Conclusion

5.1. Summary

Mining educational data is far from conclusive, yet it has been evolving and growing continuously. Using the appropriate tools and the research lines in this area are not only going to help students and instructors but also the other stakeholders/users and that impact has been extended to parents, society, and the public in general. The main focus of this research is on how EDM can support student learning in regards to student academic performance, engagement, and intervention through predicting the academic performance of students.

Student academic performance prediction has been a research topic of a significant value for many years because students and institutions can gain findings and insights uncovered from learning data. Institutions can gain from it by trying to enhance the quality of academic services and resources made available to their students in order to increase the rates of students who can progress with their study and be successful in completing their programs or courses of study. In addition, findings or insights can be deployed to deliver solutions, suggestions, or advices to students in order to improve their performance in the future.

5.2. Future Work

Since random forest has been the best model implemented and it gave an accuracy of 85%, it would be advisable to try adopting advanced methods for classification models like genetic algorithms and see if they can further improve the accuracy/performance of the model. Furthermore, with new data added, models can be tested again and see if there is any improvement in accuracy. Once a model has shown better performance, it is easy to adopt and use it for deployment on Shiny and Azure Machine Learning Studio.

REFERENCES

[1] Mandinach, E. B.: A perfect time for data use: Using data-driven decision making to inform practice. *Educational Psychologist*, **47**(2), (2012), 71–85.

[2] Slade, S. and Prinsloo, P.: Learning analytics: Ethical issues and dilemmas. *American Behavioral Scientist*, **57**(10), (2013), 1510–1529, URL <Go to ISI>://WOS:000324103200009.

[3] Picciano, A.: The evolution of big data and learning analytics in american higher education. *Journal of Asynchronous Learning Network*, **16**.

[4] Zeide, E.: The structural consequences of big data-driven education. *Big Data*, **5**(2), (2017), 164–172, URL <Go to ISI>://WOS:000403939100008.

[5] Patwa, N., Seetharaman, A., Sreekumar, K., and Phani, S.: Learning analytics: Enhancing the quality of higher education. *Research Journal of Economics*, **2**(2).

[6] Roberts, L. D., Howell, J. A., Seaman, K., and Gibson, D. C.: Student attitudes toward learning analytics in higher education: "the fitbit version of the learning world". *Frontiers in psychology*, **7**, (2016), 1959–1959.

[7] Pena-Ayala, A.: Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, **41**(4), (2014), 1432–1462, URL <Go to ISI>://WOS:000330158700045.

[8] Romero, C. and Ventura, S.: Educational data mining: A review of the state of the art. *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, **40**(6), (2010), 601–618, URL <Go to ISI>://WOS:000283447800001.

[9] Khan, A. and Ghosh, S. K.: Data mining based analysis to explore the effect of teaching on student performance. *Education and Information Technologies*, **23**(4), (2018), 1677–1697.

[10] Dutt, A., Ismail, M. A., and Herawan, T.: A systematic review on educational data mining. *IEEE Access*, **5**, (2017), 15991–16005.

[11] Baker, R. S.: Educational data mining: An advance for intelligent systems in education. *IEEE Intelligent Systems*, **29**(3), (2014), 78–82, URL <Go to ISI>://WOS:000341575700014.

[12] Lopez, S. L. S., Redondo, R. P. D., and Vilas, A. F.: Predicting students' grade based on social and content interactions. *International Journal of Engineering Education*, **34**(3), (2018), 940–952, URL <Go to ISI>://WOS:000443168300010.

[13] Nasiri, M., Minaei, B., and Vafaei, F.: Predicting gpa and academic dismissal in lms using educational data mining: A case mining. In *3rd International Conference on E-Learning and E-Teaching (ICELET)*, IEEE International Conference on E-Learning and E-Teaching, IEEE Computer Soc, LOS ALAMITOS, ISBN 978-1-4673-0958-5; 978-1-4673-0956-1, 2012, pp. 53–58, URL <Go to ISI>://WOS:000310432500008.

[14] Asif, R., Merceron, A., Ali, S. A., and Haider, N. G.: Analyzing undergraduate students' performance using educational data mining. *Computers and Education*, **113**, (2017), 177–194, URL `<Go to ISI>://WOS:000406728400013`.

[15] Costa, E. B., Fonseca, B., Santana, M. A., de Araujo, F. F., and Rego, J.: Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, **73**, (2017), 247–256, URL `<Go to ISI>://WOS:000403625400025`.

[16] Burgos, C., Campanario, M. L., de la Pena, D., Lara, J. A., Lizcano, D., and Martinez, M. A.: Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Computers and Electrical Engineering*, **66**, (2018), 541–556, URL `<Go to ISI>://WOS:000429760300041`.

[17] Iam-On, N. and Boongoen, T.: Generating descriptive model for student dropout: a review of clustering approach. *Human-Centric Computing and Information Sciences*, **7**, (2017), 24, URL `<Go to ISI>://WOS:000396492500001`.

[18] Campagni, R., Merlini, D., and Verri, M. C.: University student progressions and first year behaviour. *Proceedings of the 9th International Conference on Computer Supported Education (Csedu), Vol 2*, pp. 46–56, URL `<Go to ISI>://WOS:000444908800004`.

[19] Najera, A. B. U., de la Calleja, J., and Medina, M. A.: Associating students and teachers for tutoring in higher education using clustering and data mining. *Computer Applications in Engineering Education*, **25**(5), (2017), 823–832, URL `<Go to ISI>://WOS:000410722900013`.

[20] Ashok, M. V. and Apoorva, A.: Clustering proficient students using data mining approach. *Advances in Computing and Data Sciences, Icacds 2016*, **721**, (2017), 70–80, URL `<Go to ISI>://WOS:000434872100008`.

[21] Alharbi, Z., Cornford, J., Dolder, L., and Iglesia, B. D. L.: Using data mining techniques to predict students at risk of poor performance. In *2016 SAI Computing Conference (SAI)*, pp. 523–531.

[22] Khan, A. and Ghosh, S. K.: Analysing the impact of poor teaching on student performance. *Proceedings of 2016 Ieee International Conference on Teaching, Assessment, and Learning for Engineering (Tale)*, pp. 169–175, URL `<Go to ISI>://WOS:000400475400029`.

[23] Parkavi, A. and Lakshmi, K.: Pattern analysis of blooms knowledge level students performance using association rule mining. *2017 Ieee International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (Icstm)*, pp. 90–93, URL `<Go to ISI>://WOS:000426986800016`.

[24] Jayanthi, M. A., Kumar, R. L., and Swathi, S.: Investigation on association of self-esteem and students' performance in academics. *International Journal of Grid and Utility Computing*, **9**(3), (2018), 211–219, URL `<Go to ISI>://WOS:000441309900001`.

[25] Gomez-Aguilar, D. A., Garcia-Penalvo, F. J., and Theron, R.: Visual analytics in e-learning. *Profesional De La Informacion*, **23**(3), (2014), 236–245, URL <Go to ISI>://WOS:000339037000003.

[26] Lotsari, E., Verykios, V. S., Panagiotakopoulos, C., and Kalles, D.: A learning analytics methodology for student profiling. *Artificial Intelligence: Methods and Applications*, **8445**, (2014), 300–312, URL <Go to ISI>://WOS:000352632400024.

[27] Saqr, M., Fors, U., Tedre, M., and Nouri, J.: How social network analysis can be used to monitor online collaborative learning and guide an informed intervention. *Plos One*, **13**(3), (2018), 22, URL <Go to ISI>://WOS:000428093900105.

[28] Okoye, K., Tawil, A. R. H., Naeem, U., Bashroush, R., and Lamine, E.: A semantic rule-based approach supported by process mining for personalised adaptive learning. *5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / the 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare / Affiliated Workshops*, **37**, (2014), 203–+, URL <Go to ISI>://WOS:000349985800025.

[29] Smirnova, E. V., Samarev, R. S., and Willmot, P.: New technology for programming teaching: Process mining usage. In *7th International Conference on Education and New Learning Technologies (EDULEARN)*, EDULEARN Proceedings, IATED, VALENICA, ISBN 978-84-606-8243-1, 2015, pp. 7330–7335, URL <Go to ISI>://WOS:000376685707055.

[30] Sedrakyan, G., De Weerdt, J., and Snoeck, M.: Process-mining enabled feedback: ”tell me what i did wrong” vs. ”tell me how to do it right”. *Computers in Human Behavior*, **57**, (2016), 352–376, URL <Go to ISI>://WOS:000370457800041.

[31] He, W.: Examining students' online interaction in a live video streaming environment using data mining and text mining. *Computers in Human Behavior*, **29**(1), (2013), 90–102, URL <Go to ISI>://WOS:000312684400013.

[32] Koufakou, A., Gosselin, J., and Guo, D. H.: Using data mining to extract knowledge from student evaluation comments in undergraduate courses. In *International Joint Conference on Neural Networks (IJCNN)*, IEEE International Joint Conference on Neural Networks (IJCNN), IEEE, NEW YORK, ISBN 978-1-5090-0619-9, 2016, pp. 3138–3142, URL <Go to ISI>://WOS:000399925503046.

[33] Krishnaveni, K. S., Pai, R. R., and Iyer, V.: Faculty rating system based on student feedbacks using sentimental analysis. *2017 International Conference on Advances in Computing, Communications and Informatics (Icacci)*, pp. 1648–1653, URL <Go to ISI>://WOS:000427645500273.

[34] Rajeswari, A. M., Sridevi, M. R., and Deisy, C.: Outliers detection on educational data using fuzzy association rule mining. In *Int. Conf. on Adv. in Comp., Comm., and Inf. Sci. (ACCIS-14)*.

[35] Weng, C.-H.: Mining fuzzy specific rare itemsets for education data. *Knowl.-Based Syst.*, **24**, (2011), 697–708.

[36] Oeda, S. and Hashimoto, G.: Log-data clustering analysis for dropout prediction in beginner programming classes. In *Procedia Computer Science*, vol. 112, pp. 614–621.

[37] Amrieh, E., Hamtini, T., and Aljarah, I.: Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, **9**, (2016), 119–136.

[38] Heale, R. and Twycross, A.: Validity and reliability in quantitative studies. *Evidence Based Nursing*, **18**(3), (2015), 66.

EVALUATION OF PYTHON BASED NLP FRAMEWORKS FOR TEMPLATE BASED AUTOMATIC QUESTION GENERATION

Walelign Tewabe Sewunetie
University of Miskolc, Hungary
Department of Information Engineering
sewunetie@ait.iit.uni-miskolc.hu

László Kovács
University of Miskolc, Hungary
Department of Information Engineering
kovacs@iit.uni-miskolc.hu

[Received . . . and accepted . . .]

Abstract. Automatic question generation techniques emerged as a solution to the challenges facing test developers in the development of smart e-tutoring systems. The current challenge in selecting the available developer tools is depend on several aspects, including the kind and source of text, where the level, formal or informal, may influence the performance of such tools. This tool, popular packages for NLP: NLTK, spaCy, TextBlob, and CoreNLP.
Our experiences show that spaCy is several times faster than others in tokenization, tagging and parsing. It has also the best feature set of neural network models and of entity recognition methods. Based on our test results spaCy would be an optimal choice for the implementation of template based automatic question generation. The downside of spaCy is the limited number of supported languages. The choice which NLP package to choose depends on the specific problem you have to solve.

Keywords: Python, NLP Frameworks, Template Based Question Generation, spaCy, NLP

1. Introduction

Natural Language Processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human, in particular how to program computers to process and analyze large amounts of natural language data. It mainly concerns about teaching machines how to understand human languages and extract meaning from text [1].

Natural language processing is a computer process that requires superior knowledge of mathematics, machine learning, and linguistics. Now, developers can use ready-made tools that simplify text preprocessing so that they can concentrate on building machine learning models [1]. Google, Amazon, or Facebook are pouring millions of dollars into NLP line of research to power their chatbots, virtual assistants, recommendation engines, and other solutions powered by machine learning. NLP relies on advanced computational skills, developers would like to use the best available tools for creating services that can handle natural languages.

There are many things about Python that make it a really good programming language choice for an NLP projects. The simple syntax and transparent semantics of this language make it an excellent choice for complex projects like NLP tasks. Moreover, developers can enjoy excellent support for integration with other languages and tools that come in handy for techniques like machine learning.

Python provides developers with an extensive collection of NLP tools and libraries that enable developers to handle a great number of NLP-related tasks such as document classification, topic modeling, part-of-speech (POS) tagging, word vectors, and sentiment analysis. AQG is characterized as the task of generating syntactically sound, semantically correct, and appropriate questions from multiple input formats such as text, a structured database, or a knowledge base.

Asking assessment questions is an essential feature of advanced learning technologies such as smart tutoring systems, game-based learning environments and inquiry-based environments [2]. A general human technique for generating questions is to thoroughly read the article setting up an internal model of information and then generating questions accordingly. In the case of automated question generation (AQG) the engine generates the questions automatically from the available text documents. Many AQG systems are used in educational applications, such as skill development assessment and knowledge assessment. In the Extended ITS Architecture [3] the question generation module is using intuitionistic logic for evaluation of the generated questions. The field of AQG is an important research area that can be useful in intelligent tutoring systems, dialog systems, educational technology, educational games, e.t.c [4].

The main goal of the study is to compare, to test and to analyze different available Python based NLP frameworks for template based question generation. Template-based QG is a baseline which utilizes templates created by experts of human extracted from training set and then generates questions by filling the particular templates with certain topic entities.

2. Extensions of the article Evaluation of Python-based NLP Frameworks

2.1. Question Generation from Databases

One approach for AQG is to use a database, like relational database for the input source. The relational database [6] has the benefit that it contains a strict structure to store information and data. This structure enables to determine the meaning, semantic role of the different data items. In this case, the schema may refer to the different semantic components using references to the column names. On the other hand, to generate the NL sentences, the framework requires an NLP module to transform the lemma forms into the corresponding inflected forms.

The formal model of the database oriented AQG system can be given as follows.

- $D : \{T_1, T_1, ..., T_n\}$: input database
- $T_i : T_i(m_{i1}, m_{i2}, ..., m_{im})$: table schema containing columns
- $S : \{(S_i, Q_i)\}$: set of QA schemas
- $S_i = w_1, w_2, ..., w_m$: query schema, where w_i denotes either a NL word or a reference to a table column of the form $T_i.m_j$
- Q_i : *select ...from ...where ...*, the SQL query to yield the answer, where the SQL query can contain references to the symbol parameters used in the query schema.

For the tests, we have implemented a background database in sqlite. The database schema contains the following tables.

In Table 1, we can find examples for the generated QA schema with references to the implemented database. To implement this system we have used SQLite tools to create a relational database and python for template-based question answering systems.

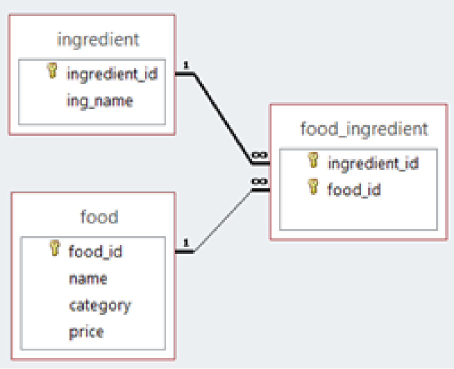


Figure 1. Restaurant food ingredient database schema

No	Question/Query Template	Sample
1	S: What is the active ingredient in @food.name? Q: Select i.ing_name From ingredient i inner join food_ingredient g on i.ingredient_id = g.ingredient_id inner join food f on f.food.id = g.food_id where f.name = @food.name	S: What is the active ingredient in potato? A: vitamin C, potassium, phosphorus and magnesium
2	S: What is the price of @food.name? Q: select price from food where name = @food.name	S: What is the price of potato? A: 20
3	S: What is the category of @food.name? Q: select category from food where name = @food.name	S: What is the category of orange? A: fruit
4	S: What kind of food have ingredient @ing.name? Q: Select f.name From ingredient i inner join food_ingredient g on i.ingredient_id = g.ingredient_id inner join food f on f.food.id = g.food_id where i.name = @ing.name	S: What vegetable have more protein? A: Edamame, Lentils

Table 1. Sample question templates with answers

As the examples show the accuracy of database oriented question generation is very high but it needs to create a template for all possible ways of questions. Thus, in case of large domain it is more challenging and time taking to construct all the required templates. In future, we will extend this work for automatized schema generation for databases. Here below is a sample python code that we have used for a template-based question answering system with database support.

```
Rules = [("What is the active ingredient in @food.name",
          "Select i.ing_name From ingredient i inner join
          food_ingredient g on i.ingredient_id = g.ingredient_id
          inner join food f on f.food.id = g.food_id where f.name
          = @food.name"),
          ("What is the price of @food.name","select price from
          food where name = '@food.name'")

        ]


def qa_process (qid):
    conn = sqlite3.connect('../Python_Cube/aqgtest')
    cur = conn.cursor()

    qid = 1
    qry = Rules[qid][0]
    qrys = qry.split("@")
    qrys2 = qrys[1].split(" ")[0]
    qryss = qrys2.split(".")

    sql = "select " + qryss[1] + " from " + qryss[0]
    result = conn.execute(sql);
    for rec in result:
        break
    qtext = qry.replace("@"+qrys2,rec[0])+"?"
    #print ("Q:", qtext)
    qry = Rules[qid][1]
    sql = qry.replace ("@"+qrys2,rec[0])
    #print (sql)
    result = conn.execute(sql);
    for rec in result:
        break
    atext = str(rec[0])
    #print ("A:", atext)
    return (qtext,atext)
```

2.2. Question Generation from Free Text

NLP plays a critical role in many intelligent applications such as automated chat bots, article summarizers, multi-lingual translation and opinion identification from data. Every industry which exploits NLP to make sense of unstructured text data, not just demands accuracy, but also swiftness in obtaining results [11]. Some of the tasks in NLP are text classification, entity detection, machine translation, question answering, and concept identification. Python is a top developing software that can handle natural languages in the context of artificial intelligence. For the implementation of Template Based Question Generation the researchers' analyzed different python based NLP frameworks.

In a research work Nguyen-thinh le et al [6] use extracted key concepts to generate questions and determine the types of questions to be generated. They use the domains of energy and economy topic to select the sentences and question. The author presents nouns and noun phrases first extracted from a discussion topic and replace by X placeholder. The template displayed in Table 1 shows that question templates are filled with the noun phrase "nuclear energy" and result in some questions. As we have seen on the template below questions are more dependent on domain and topic.

The authors used an improved NER spaCy which is capable of labeling more entity types, including money, dates/times, etc to generate questions containing the question word Why, How much, to what extent etc, [8].

In the work [8] the template creation focuses on the events (actions, happenings) and existents (characters, settings). The questions in the templates ask about the subject, the predicate, and the object of the events and existents. After removal the errors, they created 19 improved templates under 6 categories that are included in the system [8].

In David's [9] thesis report he explored semantics-based templates that uses Semantic Role Labeling (SRL) in conjunction with generic and domain-specific scope for self-directed learning. According to his report the questions that are generated are not answerable from the original sentence, they were judged answerable from the source document in our evaluation. The ability to generate questions that require the learner to consult other parts of the text is due to the flexibility of the templates.

In this study the authors have used spaCy libraries for POS tagging and this information is used to identify the potential content for the template of

Type	Question
Definition	What is @X? What do you have in mind when you think about @X? What does @X remind you of?
Feature/Property	What are the properties of @X? What are the (opposite)-problems of @X? What features does @X have?
Example	What is an example of @X
Verification	Is there any problem with the arguments about @X?
Judgment	What do you like when you think of or hear about @X
Interpretation	How can @X be used today?
Expectation	How will @X be in the future, based on the way it is now?
Quantification	How many sub-topics did you partners talk about? Which sub-topics do you partners focus on?
Concept Comparison	What is the difference or relations between these sub-topics?

Table 2. Question Templates Proposed for AQG [6]

As recently as 12,500 years ago, the Earth was in the midst of a glacial age referred to as the Last Ice Age.
T: How would you describe [A2-Ipp misc]? Q: How would you describe the Last Ice Age?
T: Summarize the influence of [A1-lp !comma !nv] on the environment. Q: Summarize the influence of a glacial age on the environment.
T: What caused [A2-Ipp nv misc]? ## [A0 null] Q: What caused the Last Ice Age?

Table 3. Sample templates and questions [9]

questions [10]. A part of the speech tagger are used to encode necessary information. In order to decide the type of questions that can be produced from this sentence, verb, object and preposition will be categorized on the basis of the subject.

3. Comparative analysis of NLP Libraries in Python for Template Based Question Generation

The most common tools and libraries that created to solve NLP problems are Natural Language Toolkit (NLTK), spaCy, TextBlob, and CoreNLP. The NLTK , for English written in the Python programming language, is a suite of libraries and programs for symbolic and statistical NLP [13]. It can include various datasets in multiple languages that can be deployed depending on the features you need. Stanford CoreNLP is a community that created core NLP components such as Tokenization, Sentence Recognition, POS Tagging, NER, Entity Linking and Training Annotation, etc [14]. The most distinctive characteristic of the Stanford NLP Group is its successful integration of advanced and deep linguistic modeling and data processing with innovative probabilistic approaches to NLP, machine learning, and deep learning. The comparison of the features offered by spaCy, NLTK, TextBlob and Stanford CoreNLP in table 4 shows that spaCy is an advanced modern NLP library. spaCy have pre-trained NLP models capable of performing the most common NLP tasks, such as tokenization, POS tagging, NER recognition, lemmatization and word vector transformation [15]. TextBlob is a Python library which offers a simple API for accessing its methods and carrying out basic NLP tasks. It offers a simple API for diving into specific NLP tasks such as part-of - speech tagging, extraction of the noun phrase, interpretation of emotions, classification, translation and more [16].

Table 5 shows the comparison of per-document processing time of various spaCy functionalities against other NLP libraries. We show both absolute timings (ms) and relative performance (normalized) to spaCy [17].

Reviewed papers confirmed that spaCy offers the fastest syntactic parser in the world and that its accuracy is within 1% of the best available. The few systems that are more accurate are 20x slower or more [17].

Spacy is very powerful and industrial strength package for almost all natural language processing tasks. The following figure shows the comparison of spaCy with CoreNLP and NLTK based on accuracy for entity extraction.
Spacy consists of a fast entity recognition model which is capable of identifying entity phrases from the document. Entities can be of different types, such as person, location, organization, dates, numerals, etc. These entities can be accessed through “.ents” property. According to the research work [18] and we have also observe that spaCy is easy to use, provides the best overall performance compared to Stanford CoreNLP Suite, Google’s SyntaxNet, and NLTK Python library.

Feature	spaCy	NLTK	Stanford CoreNLP	TextBlob
Programming Language	✓			
Easy Installation	✓	✓	✓	
Neural Network Models	✓	✓		
Integrated word vectors	✓			
Multi language support	✓	✓	✓	
Tokenization	✓	✓	✓	
Part of Speech Tagging	✓	✓	✓	
Sentence segmentation	✓	✓		
Dependency parsing	✓	✓	✓	
Entity recognition	✓	✓		
Stemming	✓	✓	✓	
Lemmatization	✓	✓	✓	

Table 4. Comparison of the functionalities offered by spaCy, NLTK, TextBlob and Stanford CoreNLP

System	Tokenize	Tagging	Parsing	Tokenize	Tag	Parse
spaCy	0.2 ms	1 ms	19 ms	1x	1x	1x
coreNLP	018 ms	10 ms	49 ms	0.9x	10x	2.6x
ZPar	1 ms	8 ms	850 ms	5x	8x	44.7x
NLTK	4 ms	443 ms	n/a	20x	443x	n/a

Table 5. Compare the per-document processing time of various spaCy functionalities against other NLP libraries

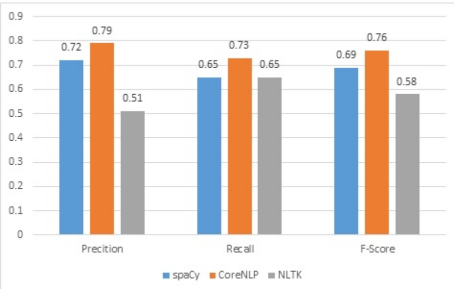


Figure 2. Accuracy for entity extraction

One of the most powerful feature of spacy is the extremely fast and accurate syntactic dependency parser which can be accessed via lightweight API. The parser can also be used for sentence boundary detection and phrase chunking.

The relations can be accessed by the properties ".children", ".root", ".ancestor" etc[12].

4. Feature Level Comparison

The two significant libraries used in NLP are NLTK and spaCy. There are substantial differences between them, which are as follows: NLTK provides a plethora of algorithms to choose from for a particular problem which is boon for a researcher but a bane for a developer. Whereas, spaCy keeps the best algorithm for a problem in its toolkit and keep it updated as state of the art improves.

NLTK is a string processing library and it takes strings as input and returns strings or lists of strings as output. Whereas, spaCy uses object-oriented approach. When we parse a text, spaCy returns document object whose words and sentences are objects themselves. spaCy has support for word vectors whereas NLTK does not. As spaCy uses the latest and best algorithms, its performance is usually good as compared to NLTK. As we can see below, in word tokenization and POS-tagging spaCy performs better, but in sentence tokenization, NLTK outperforms spaCy. Its poor performance in sentence tokenization is a result of differing approaches: NLTK attempts to split the text into sentences. In contrast, spaCy constructs a syntactic tree for each sentence, a more robust method that yields much more information about the text.

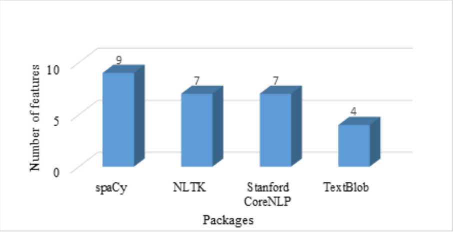


Figure 3. Number of features offered by spaCy, NLTK, Stanford CoreNLP and TextBlob

The most popular NLP tools available in Python, spaCy supports 9 features out of 10. In our observation spaCy have fast processing speed in 3 major key functionalities Tokenization, POS Tagging, Entity Extraction.

SpaCy, on the other hand, is the way to go for app developers. While NLTK provides access to many algorithms to get something done, spaCy provides the best way to do it. It provides the fastest and most accurate syntactic analysis of any NLP library released to date. It also offers access to larger word vectors that are easier to customize. For an app builder mindset that prioritizes getting features done, spaCy would be the better choice. Both NLTK and spaCy offer great options when you need to build an NLP system. As we have seen, however, spaCy is the right tool to use in a production environment.

5. Conclusion

In this article, we compared some features of several popular NLP libraries. While most of them provide tools for overlapping tasks, some use unique approaches for specific problems. Definitely, the most popular packages for NLP today are NLTK and spaCy. In our opinion, the difference between them lies in the general philosophy of the approach to solving problems.

You can use it to try different methods and algorithms, combine them, etc. spaCy, instead, provides one out-of-box solution for each problem. Also, spaCy is several times faster than NLTK. Despite the popularity of these two libraries, there are many different options, and the choice which NLP package to choose depends on the specific problem you have to solve. spaCy would be an optimal choice for template based question generation.

Acknowledgements

The research reported here was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University – Innovative Knowledge City – Institutional development of the University of Miskolc aiming at intelligent specialization" project implemented in the framework of the Széchenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] Dominik Kozaczko, *Best python natural language processing nlp libraries*, [Online]. Available: https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp-libraries/,2018.

[2] K. E. a. P. P. e. Boyer: Proceedings of QG2010. *The Third Workshop on Question Generation, in Pittsburgh*, questiongeneration.org, 2010.

[3] Walelign T.S et al: *The development and analysis of extended of architecture model for intelligent tutoring systems*, Gradus ISSN 2064-8014 , vol. 6, no. 4, pp. 128-138, 2019.

[4] J. P. a. I. S. Dhaval Swali:, *Automatic Question Generation from Paragraph*, International Journal of Advance Engineering and Research Development, vol. 3, no. 12, pp. 73-78, 2016.

[5] M. H. N. A. Smith:, *Good Question! Statistical Ranking for Question Generation, in Human Language Technologies:* Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, , Los Angeles, California, USA, June 2-4, 2010.

[6] N.-T. Le and N. Pinkwart, *Evaluation of a question generation approach using semantic web for supporting argumentation*, Research and Practice in Technology Enhanced Learning , vol. 10, no. 3, p. 19, June 2015.

[7] , G. Keswani,*AutoQuest (An Intelligent Automatic Question Paper Generator System)*, Abdul Kalam Technology University, Lucknow, 2018-19.

[8] K. Mhartre, *Question Generation using NLP*, International Journal of Scientific Research & Engineering Trends, vol. 5, no. 2, pp. 394-397, 2019.

[9] E. L. Fasya, *Automatic question generation for virtual humans*,Enschede, The Netherlands, August 2017.

[10] D. Lindberg, *Automatic question generation from text for self directed learning*,Simon Fraser university, Canada, 2013.

[11] Mandasari Yani: *Follow-up question generation*, University of Twente M.Sc Thesis, 2019.

[12] S BANSAL, *Natural Language Processing Made Easy using spaCy (in Python)*,[Online]. Available: https: //www.analyticsvidhya. com/blog/2017/04/ natural-language-processing-made-easy-using-spacy- [Accessed 19 10 2020].

[13] Awesome Python, *Natural Language Processing packages and projects*, https: //python.libhunt.com/categories/169-natural-language-processing,[Accessed 19 10 2020].

[14] Manning and Christopher D. at al, *The Stanford CoreNLP Natural Language Processing Toolkit*, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60, 2014.

[15] Online,*Python PoS Tagging and Lemmatization using spaCy*, Available: https://www.geeksforgeeks.org/python-pos-tagging-and-lemmatization-using-spacy/,[Accessed 19 10 2020].

[16] S. Loria, *TextBlob: Simplified Text Processing*, [Online]. Available: https://textblob.readthedocs.io/en/dev/. [Accessed 25 07 2020].

[17] spaCy, *Facts & Figures*, [Online]. Available: https://spacy.io/usage/facts-figures. [Accessed 16 4 2020].

[18] F. N. A. Al Omran and C. Treude, *Choosing an NLP Library for Analyzing Software Documentation A Systematic Literature Review and a Series of Experiments*,IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) DOI: 10.1109/MSR.2017.42, pp. 187 - 197, 20-21 May 2017.

[19] M. H. N. A. Smith, *Good Question! Statistical Ranking for Question Generation, in Human Language Technologies*: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, , Los Angeles, California, USA, June 2-4, 2010.

Notes for Contributors

related to the papers to be submitted
for Production Systems and Information Engineering

Aims and scope

The aim of the journal is to publish high quality research papers connected with both production systems and information engineering at the same time. Special emphasis is given to articles on the theoretical models and methods, as well as practical applications of discrete production processes including new (or partially new) software tools. Using a new term proposed in special literature in the nineties, the main profile of this journal is Production Information Engineering.

Frequency of the journal

One volume a year (80–200 pages per volume) is planned.

Submission of manuscript

Submission of a manuscript implies that the paper has not been published, nor is being considered for publication elsewhere. All papers should be written in English. Two copies of the manuscript should be submitted on pages of A4 size.

Each manuscript should be provided with an English Abstract of about 50–100 words, reporting concisely on the objective and the results of the paper. The English Abstract is to be followed by the keywords.

References should be grouped at the end of the paper in numerical order of appearance. Author's name(s) and initials, paper titles, journal name, volume, issue, year and page numbers should be given for all journals referenced.

For the purpose of refereeing, papers should initially be submitted in hardcopy (twofold) to the secretaries. The eventual supply of an accepted-for-publication paper in its final camera-ready form will ensure more rapid publication. As regards the detailed format requirements, please consider the next homepage: http://ait.iit.uni-miskolc.hu/˜psaie.

A Short History of the Publications of the University of Miskolc

The University of Miskolc (Hungary) is an important centre of research in Central Europe. Its parent university was founded by the Empress Maria Teresia in Selmecbánya (today Banska Stiavnica, Slovakia) in 1735. After the First World War the legal predecessor of the University of Miskolc moved to Sopron (Hungary) where, in 1929, it started the series of university publications with the title Publications of the Mining and Metallurgical Division of the Hungarian Academy of Mining and Forestry Engineering (Volumes I–VI). From 1934 to 1947 the Institution bad the name Faculty of Mining, Metallurgical and Forestry Engineering of the József Nádor University of Technology and Economics Sciences at Sopron. Accordingly, the publications were given the title Publications of the Mining and Metallurgical Engineering Division (Volumes VII–XVI). For the last volume before 1950 due to a further change in the name of the Institution Technical University, Faculties of Mining, Metallurgical and Forestry Engineering, Publications of the Mining and Metallurgical Divisions was the title. For some years after 1950 the Publications were temporarily suspended.

After the foundation of the Faculty of Mechanical Engineering in Miskolc in 1949 and the movement of the Sopron Mining and Metallurgical Faculties to Miskolc the Publications restarted with the general title Publications of the Technical University of Heavy industry in 1955. Four new series Series A (Mining), Series B (Metallurgy), Series C (Machinery) and Series D (Natural Sciences) were founded in 1976. These came out both in foreign languages (English, German and Russian) and in Hungarian.

After the foundation of the Faculty of Mechanical Engineering in Miskolc in 1949 and the movement of the Sopron Mining and Metallurgical Faculties to Miskolc the Publications restarted with the general title Publications of the Technical University of Heavy industry in 1955. Four new series Series A (Mining), Series B (Metallurgy), Series C (Machinery) and Series D (Natural Sciences) were founded in 1976. These came out both in foreign languages (English, German and Russian) and in Hungarian. In 1990, right after the foundation of some new faculties, the university was renamed the University of Miskolc. At the same time the structure of the Publications was reorganized so that it could follow the faculty structure. Accordingly, three new series were established: Series E (Legal Sciences), Series F (Economic Sciences), and Series G (Humanities and Social Sciences). The seven series are constituted by some periodicals and publications, which come out with various frequencies.

PRODUCTION SYSTEMS AND
INFORMATION ENGINEERING
Volume 9 (2020)

CONTENTS