DOI: 10.24427/978-83-66391-87-1_06

Viktoriia BORTNIKOVA[1], Vladyslav YEVSIEIEV[1], Iryna BOTSMAN[1],
Igor NEVLIUDOV[1], Kostiantyn KOLESNYK[2], Nazariy JAWORSKI[2]

6. QUERIES CLASSIFICATION USING MACHINE LEARNING FOR IMPLEMENTATION IN INTELLIGENT MANUFACTURING

Today, information retrieval systems have plays an important role in intellectual manufacturing. Such systems would provide a speed with large volumes of data, system speed and etc. More roles in such system is search queries. The search queries are arrays of digital text information that can be big data, coming up to several hundred billion gigabytes or higher. In order to increase the speed of operation with such data in information retrieval systems, it is necessary to classify them.

For task it is known: production section, 1000 search queries of arbitrary form and content and 5 search queries categories (error message, production section, the equipment state, sensors search, and system parameters).

So here is the task lay down of the information retrieval systems speed increasing in manufacturing at the expense search queries classification by machine learning methods.

6.1. RESEARCH OBJECTIVE

Here is the task lay down of the information retrieval systems speed increasing in manufacturing at the expense of the search queries classification by machine learning methods. For this task, it is known:

- production section,
- 1000 search queries of arbitrary form and content,
- search queries categories: error message, production section, the equipment state, sensor search, system parameters,
- only one value for each category can be assigned to each search query, and a set of possible values for each category is known beforehand,

[1] Kharkiv National University of Radio Electronics, Ukraine
[2] Lviv Polytechnic National University, Ukraine

- the search queries need to be analyzed and categorized into several unrelated categories, and the search query content should be defined as one of them,
- when classifying search queries, categories are defined in advance, with clustering they are not specified, and even information about their number may not be available. It is necessary to perform mathematical statement of the search queries classification task. To do this, consider the existing formulations of the classification problem.

Formally, the classification problem is the next: an array of text search queries $T = \{t_1,\ t_2,\ \ldots,\ t_i\}$ and an array of possible classes $C = \{c_1,\ c_2,\ \ldots,\ c_i\}$ are set. There is an unknown target dependency – a transform image $f: T \times C \to \{0; 1\}$, that is set

$$f(t_i, c_i) = \begin{cases} 0, \text{if } t_i \notin c_j, \\ 1, \text{if } t_i \in c_j. \end{cases} \quad (6.1)$$

It is necessary to form a classifier $f'(t_i, c_j)$ that is as close as possible to $f(t_i, c_j)$. With such statement of search queries classification task, there is no known additional information about the classes and the search queries text other than those that can be derived from the search query itself. The classification will be accurate when the resulting search query classifiers transform the image

$$f': T \times C \to \{0; 1\}.$$

The search query class will be the limit if such degree of similarity will yield

$$f': T \to [0; 1].$$

Described statement of the problem refers to the tasks of machine learning by precedents or training with a teacher [1]. In the general case, the training sample *N* is formed that is a set of search queries related by the previously unknown regularity. This sample is necessary for the classifier training and determining the values of its parameters, with which the classifier produces a better result. Next, in the system, the decisive rules will be determined, by which the search queries set division for given classes occur.

In the set task, each search request must respond only to one class $c \in C$, and then there will be unambiguous classification.

Thus, it is necessary to solve several tasks for the search queries classification:

- search query text pre-processing,
- search queries attributes identification,
- search queries attributes dimensionality decreases,
- classifier development and training by the machine learning methods,
- classification quality assessment,
- obtaining a classifier model,
- classifier testing for new data.

For the classification algorithm, choosing the particular qualities of each algorithm should be taken into account and as a result, it is necessary to conduct research. It is also necessary to resolve the issues of determining the attribute set, their number and the methods of calculating weight numbers, and also of the need to select some algorithms parameters during the training step.

In the deep learning algorithms, the classification accuracy depends on the availability of a training sample of the appropriate size, and the preparation of such sample is a very laborious process.

6.2. NORMALIZATION OF SEARCH QUERY DATA

Each search query text *T* consists of

$$T = S \cup W, \tag{6.2}$$

where S is the word's array of the search query text $S = [word_1 \ldots word_n]$, n is the word's number, and W is the set of words that do not carry semantic meaning (unions, pronouns, articles, numbers, signs, etc.).

To simplify the work with the search query texts, suppose that W can also be defined as a set S, that is

$$T = [word_1 \ldots word_n] \tag{6.3}$$

Pre-processing of the search query text is necessary before its conversion into numerical values and further work on it. First of all, the noise component must be removed from text, particularly, removal of words that do not carry a semantic meaning. As noted above, such words are unions, pronouns, articles, numbers, signs, and so on.

6.2.1. TOKENIZATION

To do this, we first need to split up the search query for words or phrases *tokens* (*tokenization*), taking into account the specifics of the search query text, i. e. *technological process* should be perceived as one or two *tokens*. To implement *tokenization*, use N-gram [2-3]. N-grams are of several types. The most common search queries when using tokenization are unigrams and bigrams. Also, there are the symbolic N-grams in which the text is not fragmented into separate words, but on the characters segments of a certain length [2-5]. A comparative analysis was conducted, the results of which are given in Table 6.1. Several variants of search query tokenization in the form of the unigrams, the bigrams and the 3-character N-grams were analyzed for the example of 3 requests: *workpiece location 8*, *technological operation 3* and *sensors status 26 in production line 2*.

As Table 6.1 shows, the best variant would be to use either unigrams or bigrams, because the character N-gram divides the search query text into unrelated letters that is difficult for further processing.

Table 6.1. Results of search queries tokenization by N-grams

Search queries	Unigrams	Bigrams	3-character N-grams
Technological operation 3	[technological, operation, 3]	[technological operation, 3]	[tec, hno, log, ica, lo, per, ati, on, 3]
workpiece location 8	[workpiece, location, 8]	[workpiece location, 8]	[wor, kpi, ece, lo, cat, ion, 8]
sensors status 26 in production line 2	[sensors, status, 26, in, production, line, 2]	[sensors status, 26 in, production line, 2]	[sen, sor, s s, tat, us, 26 in, np, rod, uct, ion, li, ne , 2]

For unigram, the search query text will be written as

$$T = [word_1, word_2, \dots, word_n]. \tag{6.4}$$

6.2.2. CALCULATE QUALITY PARAMETERS OF THE SEARCH QUERY TEXT DEFINITION

After the search query splitting (*tokenization*) into *words*, it is necessary to perform its syntactic and spelling check, as well as to determine its unmeaning and informativeness [6]. As a generalized estimate of the search query text, the next expression can be used

$$\lambda = \frac{\omega_1 \cdot E + \omega_2 \cdot O + \omega_3 \cdot V + \omega_4 \cdot P}{\sum_{j=1}^{4} \omega_j}, \tag{6.5}$$

where E is the calculated value of the search query text syntactic correctness; O is the estimated value of the search query text spelling correctness; V is the estimated value of the search query text unmeaning; P is the estimated value of the search query text informativeness; ω_i is weight coefficients that represent the significance of one or another parameter in the overall assessment. It should be noted that

$$\sum_{j=1}^{4} \omega_j = 1,$$

then (6.5) simplifies to

$$\lambda = \omega_1 \cdot E + \omega_2 \cdot O + \omega_3 \cdot V + \omega_4 \cdot P. \tag{6.6}$$

It is necessary to determine the weight coefficients for each of the parameters. This can be performed using the methods of ranking and assigning points [6]. Expertise is conducted by the experts group of 20 people who are experts in the field of automation and computer-integrated technologies and have different age categories (Table 6.2).

Weighting coefficients that represent the significance of one or another parameter in the overall assessment

$$\omega_1 = \frac{64}{200} = 0.305, \quad \omega_2 = \frac{36}{200} = 0.18,$$
$$\omega_3 = \frac{61}{200} = 0.305, \quad \omega_4 = \frac{42}{200} = 0.21. \tag{6.7}$$

The obtained values of weight coefficients can be substituted to (6.6) [7]

$$\lambda = 0.305 \cdot E + 0.18 \cdot O + 0.305 \cdot V + 0.21 \cdot P. \tag{6.8}$$

Next, we consider how it is possible to calculate quality parameters of the search query text definition.

Table 6.2. Expert polls results

Expert	Parameters important for evaluating search query text (rank)				Expert	Parameters important for evaluating search query text (rank)			
	V	*E*	*P*	*O*		*V*	*E*	*P*	*O*
1	4	3	2	1	11	3	2	1	4
2	3	2	4	1	12	3	1	4	2
3	3	2	4	1	13	3	1	4	2
4	3	1	4	2	14	3	1	4	2
5	2	1	4	3	15	2	1	4	3
6	2	4	1	3	16	4	2	3	1
7	4	2	3	1	17	3	2	4	1
8	4	1	2	3	18	2	1	4	3
9	4	1	2	3	19	2	4	1	3
10	4	3	2	1	20	3	1	4	2
Sum	33	20	28	19	Sum	28	16	33	23

The syntax check of a search query text involves checking of the search query text syntactic correctness that can be calculated as follows

$$E = 1 - \begin{cases} 2 \cdot \frac{s}{n}, \text{ if } \frac{s}{n} < 0.5, \\ 1, \text{ if } \frac{s}{n} \geq 0.5, \end{cases} \tag{6.9}$$

where s is the syntax errors number, n is the total words number in the search query.

It must be taken into account that the syntactic correctness of the search query text E must take on values from 0 to 1, the value 0 means that all words in the search query text are syntactically correct and 1 means that all words in the search query text are syntactically incorrect.

So if the number of syntax errors is greater than half of the total words number in the search query text, the text should be evaluated as syntactically incorrect. And if errors number is less than the threshold value, the estimated value will be equal to the doubled ratio of the syntax errors number to the total words number.

The search query text spell checking involves the follows spelling correctness calculation

$$O = 1 - \begin{cases} 2 \cdot \frac{o}{n}, \text{ if } \frac{o}{n} < 0.5, \\ 1, \text{ if } \frac{o}{n} \geq 0.5, \end{cases} \tag{6.10}$$

where o is number of spelling mistakes.

It should be noted that the search query text spelling correctness O must also take on values from 0 to 1. The value 0 means that all words in the search query text are spelled correctly, and 1 means that all words in the search query text are spelling mistaken. For the analysis of the search queries texts quality, it is necessary that, with the spelling mistakes number greater than half the total of words number in the search query text, the text will be evaluated as illiterate. And in case when the errors number is less than the threshold value, the assessment value will be equal to doubled ratio of spelling errors number to total words number.

Checking the search query text unmeaning involves checking for presence of meaningful expressions, phrases, words that do not carry semantic meaning in the search query text. It can be calculated as follows

$$V = 1 - \begin{cases} 2 \cdot \frac{v}{n}, \text{ if } \frac{v}{n} < 0.5, \\ 1, \text{ if } \frac{v}{n} \geq 0.5, \end{cases} \tag{6.11}$$

where v is the number of spelling mistakes.

The unmeaning of search query text V should fall in the range from 0 to 1. The value of 0 means that in the search query text completely absent of words without semantic meaning. And value of 1 means that the search query text is entirely composed of words without semantic meaning (but in practice, as a rule, such search queries texts have a random character of appearance). Thus, when the number of words that does not carry a semantic meaning in the search query text is more than half of the total words number, the text must be evaluated as the text with the maximum amount of unmeaning. When the number of words without semantic meaning in the search query text is less than the threshold value, the assessment value will be equal to the doubled ratio of the number of words without semantic meaning in the search query text to the total words number in the search query.

Checking of the search query text for informativity allows determining the search query text quality, taking into account the possible repetition of expressions, phrases, words. And in such a way, semantic meaning of the search query text can be displayed. Informative content of the search query text can be calculated as follows

$$P = \begin{cases} \frac{p}{n}, \text{ if } 0.3 < \frac{p}{n} < 0.8, \\ 1, \text{ if } \frac{v}{n} \geq 0.8, \\ 0, \text{ if } \frac{v}{n} \leq 0.3, \end{cases} \quad (6.12)$$

where p is the number of different search query words.

It should be taking into account that the information content of the search query text of the P must also take on values from 0 to 1. The value 0 corresponds to the search query text, which contain repetitive word, and 1 means that all words in the search query text differ. Having in consideration the statistics that threshold value, which corresponds to 0 in the search queries text, is rarely encountered, then (6.14) is converted as follows.

For values less than or equal to 0.3, the informative content of the search query text is equal to 0, with values greater than 0.8 informative content is 1, and for range from 0.3 to 0.8 value is unchanged.

6.2.3. STEMMING

Thus, after search query text *tokenization* and its syntactic and spelling checking, as well as unmeaning and informative determining, it is necessary to perform *stemming*. That means cutting off the words flexions and suffixes, so that the rest of the part remains the same for all grammatical word forms. The results of stemming are similar to the word root definition. But stemming algorithms are based on other principles and the results after its use often differ from the morphological *word* root. There are several variants of the stemming algorithms that differ in accuracy and productivity. They are: search by table, clipping of flexions and suffixes, lemmatization, stochastic algorithms, hybrid approach, prefixes clipping, matching search [4]–[7].

A comparative analysis of the features, advantages and disadvantages of known stemming algorithms was carried out. In this analysis, the search query text specifics of the stemming algorithm base was researched. As an example, the term *technological operation*, which is tokenized as: [*technological*, *operation*], was analyzed [8].

Based on the analysis, it can be concluded that in the future research, the algorithm of flexions and suffixes clipping will be used. It is compact and productive and will allow the words stemming in the search queries text to be realized. After search query converting into the words sequence, converting them into the attributes vector can be started.

6.3. TRAINING AND TESTING A SEARCH QUERY CLASSIFIER USING MACHINE LEARNING

Next, we set out the search query text as the list of pre-processed words T^*. Each word of the search query $word_n \in T^*, n = \overline{1, n'}$ has its own quality rating λ and weighing

coefficient W relative to the search query text $t_j \in T$. Thus, each search query text can be represented as a vector of the weighing coefficient of its words

$$\vec{t}_j = \langle W_{1j}, \dots, W_{ij} \rangle.$$

The weighing coefficient of search queries is standardized by taking into account equations (6.7), (6.9), (6.11), (6.13), and (6.15)

$$0 < W_{ij} < 1, \ \forall i,j: 0 \le i \le |T^*|, \ \ 0 \le j \le |T|.$$

Thus, a dictionary of 121 words for the search query classification can be presented in the matrix form of 121 × 1119 elements, where each line corresponds to the weighing coefficients of the meaningful search query words.

The development of a classifier for the search query classification is carried out using a neural network. One of the types of neural networks is learned networks [9]. This network type is used to formalize tasks that include the recognition of search query texts. In the process of learning the network automatically changes its parameters, such as the weighing coefficient of the layers and, if necessary, the number of hidden layers. In general, the neural network must have an input layer, hidden layers, and an output layer.

The main points of the search query classification using the neural network are as follows:

- the matrix is forming in which the values of the input layer X and the output layer Y are present; the X value is equal to the value of the processed dictionary and the value of Y corresponds to the specified value of the search query characteristics,
- selecting the required number of neural network layers,
- selecting the required number of neural network hidden layers,
- generating random values of the weighing coefficients for all neurons in the network,
- adjustment of the weighing coefficients of all neurons in the network to achieve the minimum error value,
- obtaining a trained neural network,
- evaluation of the received classifier.

The neuron input layer is the search query text converted in the dictionary form, which is considered in section 2, that is, the size of the input layer is equal to the size of the processed dictionary of 121 elements, and each neuron of the input layer is fed as a normalized number (from 0 to 1).

The output layer size for the search query classification task equals to the number of possible target values. The search query characteristics should have the following values in accordance with the appeal: system error; state of the production line; state of the technological equipment; sensors and system parameters. Then the neurons number in the output layer will be equal to 5. When training the network for each characteristic value, it is necessary to determine in advance the reference unique set of numbers that we expect to get at the output layer.

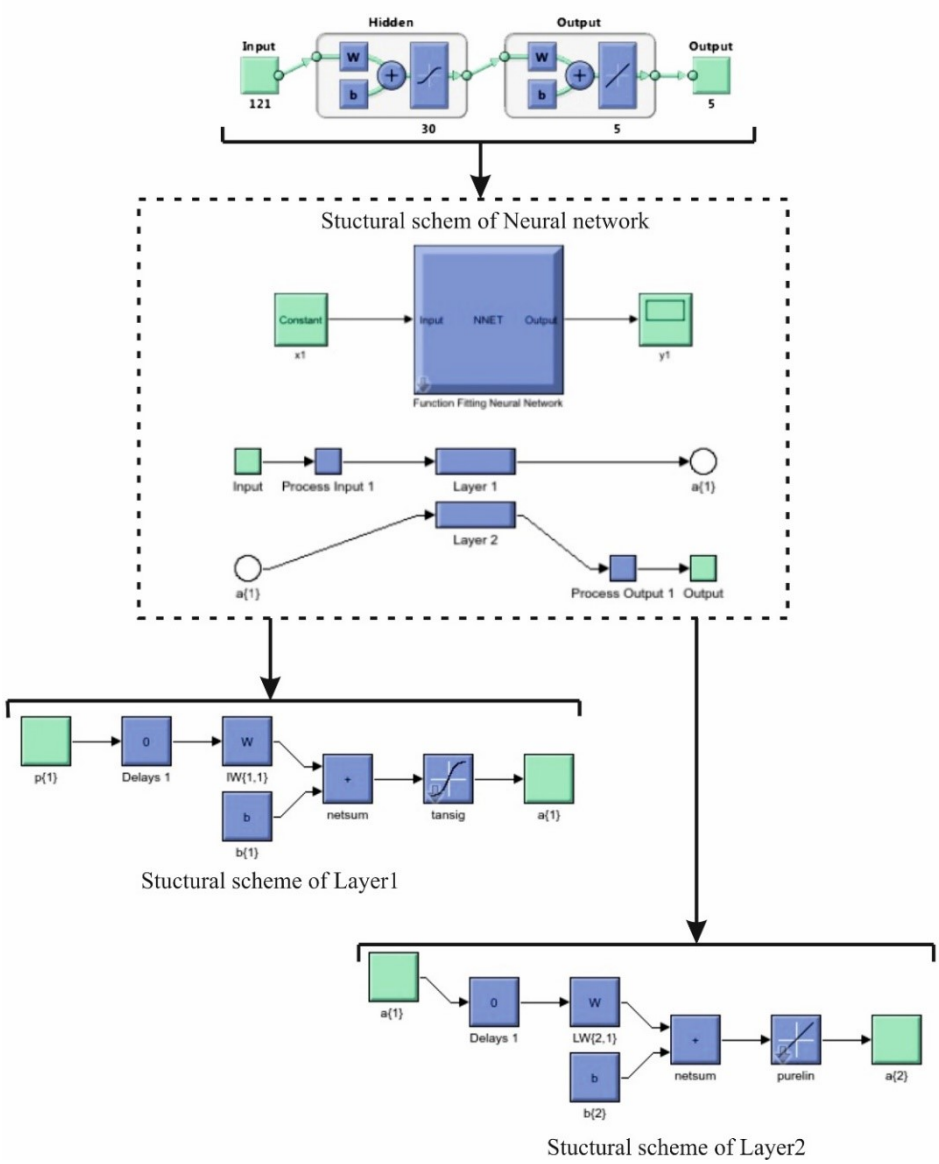

Fig. 6.1. Neural network architecture

The number of neurons in each layer needs to be determined in advance. After learning the network, each output neuron must also have a value from 0 to 1.

For training neural network, the *toolNeuralFitting* is used, in which there is a need to set the value of the inputs and targets sequence [9]. The data matrix for network learning is a pre-processed dictionary that is stored as a matrix of 121 × 1119 elements. And the vector of the output layer is *Targets*, that is, pre-processed categories (matrix values with the size of 5 × 1119 elements).

After putting in the values of the input and output layers, process of the neural network learning, which is an iterative process, begins. In the network learning process, the number and dimension of hidden layers were changed.

Experimentally determined that the best result is given by a neuron network consisting of two layers (Fig. 6.1). The hidden layer has a dimension of 30 neurons (that is approximately 1/4 of the dictionary size). And the second layer has a dimension of 5 neurons (that is 1/6 of the first layer size).

To check the network productivity, the regression can be evaluated (Fig. 6.2). The graph shows the relationship between the *Dataset* and *Outputs* (targets) for *Training*, *Validation*, and *Testing* data. For perfect fit, the data should get along the 45-degree line, where the network outputs are equal to the targets.

As can be seen from Fig. 6.2, the adjustment is good enough for all datasets, and the values of *R* in each case is 0.73 or higher. If more precise results are needed, it is possible to retrain the network with changed start weighing coefficients, that may lead to improved network after retraining, or vice versa.

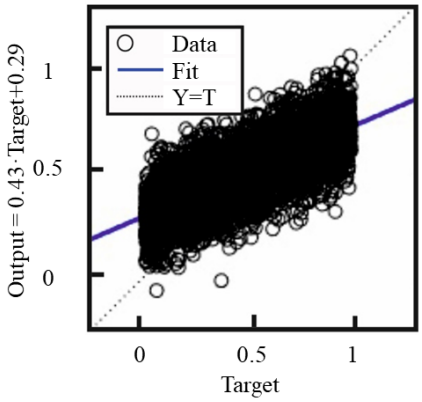

Fig. 6.2. Model productivity evaluation

We further evaluate the network productivity using the error histogram (Fig. 6.3). In Fig. 6.3, the blue columns indicate the learning data (*Training*), the green columns show check data (*Validation*), and red columns show the test data (*Testing*).

The histogram shows overshoot that are data points, where fitting is much worse than for most data. Fig. 6.3 shows that although most errors fall within the range between 0.8927 and 0.8548, there is a training point with error of 17 and verification checkpoints with errors of 12 and 13. These overshoots are also seen in the regression graph (Fig. 6.2). The first point corresponds to a point with the target of 0.5979 and is displayed at 0.3066.

The overshoots checking using the error histogram allows to determine the data quality and determine the data points that differ from the rest of the data set.

The mean-square error can also be evaluated on the validation data for successive training periods (*epoch*) and it is possible to see the changes dynamics in the learning status.

To classify the search queries in the information retrieval systems, preparation of the training sample was formed. In order to process the search queries texts, a *tokenization* of a search query for words or phrases was made initially. It was determined that the best fit for this task is using the unigram.

After *tokenization*, the syntax and spelling checking were performed, and the search query text unmeaning and informativeness were determined. To reduce the dictionary size, the *stemming* algorithm was used. Thus, the set of 1000 queries was first converted into a dictionary of 3200 elements, and as a result, the dictionary was obtained in the matrix form of 121×1119 elements.

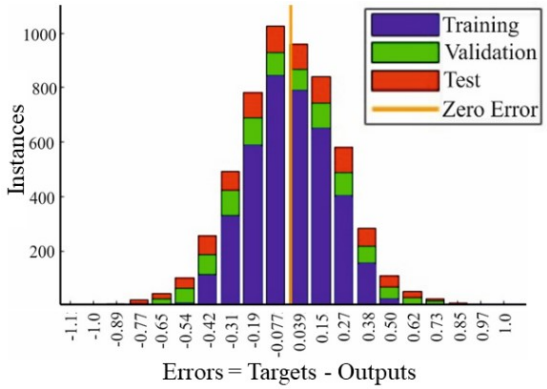

Fig. 6.3. Histogram of errors

The classifier was trained using a neural network. The classifier productivity evaluation was performed according to the error value, the noise component, the system training status and the training speed.

6.4. CONCLUSIONS

Based on the research and evaluation of the obtained learning results of the neural network, we can conclude that it is necessary to conduct further research in the direction of improving the learning outcomes of the network. To do this, it is necessary to conduct a more in-depth study aimed at improving the quality of the classifier and determining a more universal approach to solving the task of search query classification for information retrieval systems in intelligent manufacturing.

Further, the introduction of such neural network will increase the speed of information retrieval systems work by classifying search queries for 5 specified categories.

REFERENCES

[1] GUAND N., WANG X., *Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education*, IGI Global, 2012.

[2] ZHANG X., ZHAO J., LECUN Y., Character-level convolutional networks for text classification, in: *Proc. Neural Inform. Processing Systems Conf. (NIPS 2015)*, https://arxiv.org/abs/1509.01626, 2015 (accessed: 28.11.2018).

[3] JURAFSKY D., MARTIN J. H., *N-gram Language Models, Speech and Language Processing*, online https://web.stanford.edu/~jurafsky/slp3/3.pdf, (accessed: 28.11.2018).

[4] VIJAYARANI S., ILAMATHI J., NITHYA M., Preprocessing Techniques for Text Mining, *International Journal of Computer Science & Communication Networks*, vol. 5(1), p. 7-16, 2014.
[5] LABOREIRO G. et al., Tokenizing micro-blogging messages using a text classification approach, in: *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, Toronto, Canada, 2010.
[6] Universitat Pompeu Fabra, *How to write a streaming algorithm*, online https://essentia.upf.edu/documentation/extending_essentia_streaming.html, (accessed: 28.11.2018).
[7] LOVINS J. B., *Development of a Stemming Algorithm*, Mechanical Translation and Computational Linguistics, 1968.
[8] BORTNIKOVA V. et al., Search Query Classification Using Machine Learning for Information Retrieval Systems in Intelligent Manufacturing, in: *Proc. 15th International Conference ICTERI 2019 (ICTERI 2019)*, p. 460-495, 2019.
[9] KIM P., *MATLAB Deep Learning: with Machine Learning, Neural Networks and Artificial Intelligence*, Springer Science, 2017.