Search Query Classification Using Machine Learning for Information Retrieval Systems in Intelligent Manufacturing

Viktoriia Bortnikova[1][0000-0002-6215-7797], Igor Nevliudov[1][0000-0002-9837-2309],
Iryna Botsman[1][0000-0003-1110-9602] and Olena Chala[1][0000-0003-2454-3774]

[1]Department of Computer-Integrated Technologies, Automation and Mechatronics,
Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine
viktoriia.bortnikova@nure.ua,igor.nevliudov@nure.ua,
irina.botsman@nure.ua,olena.chala@nure.ua

Abstract. The problem of the information retrieval systems operation efficiency increasing in manufacturing is analysed and the general statement of research task is carried out. The training sample preparation for search queries classification was performed, during which the task of search queries classification was formulated, pre-processing of the search query texts was carried out and the dictionary was generated. For the dictionary formation, search queries tokenization and stamping were initially carried out, and then a dictionary was created in the words weight vector form based on the search query text generalized evaluation. On the basis of prepared dictionary (as the matrix of 121×1119 elements) the search queries classifier model was developed and experimentally investigated. The neural network learning was performed. In the result the network with best performance was determined, for which neurons number for the first layer is 30, and for second layer is 5. The developed neural network can be integrated into the information retrieval system in the form of a classifier in 5 categories. It will make possible to classify search queries and thereby increase the information retrieval system operation speed in manufacturing.

Keywords: Information Retrieval Systems, Search Query, Machine Learning, Intelligent Manufacturing, Neural Networks.

1 Introduction and Research Objective

The features of the intelligent manufacturing organization cause a need to process and analyze large volumes of information at different production lines. The use of Industry 4.0 technologies requires the use of information systems of various types, that are intended for automatic processing of large information volumes, search queries, information from sensors, etc. One of the promising directions for increasing the such systems speed is the recognition of search queries, what qualifies as classification tasks.

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. The incomplete determination of the output data, the high computational complexity of the decision methods, as well as the availability of specific queries, can attribute this problem to tasks with weakly structured data. For tasks of this kind, the use of machine learning methods is promising.

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 and 5 search queries categories (error message, production section, the equipment state, sensors search, and system parameters). Moreover, 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. The search queries need to be analyzed and categorized into several unrelated categories. It is necessary to perfom 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 $C = \{c_1, c_2, \ldots, c_j\}$ are set. There is an unknown target dependency – a transform image $f : T \times C \to \{0, 1\}$, that is set:

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

It is necessary to form a classifier $f'(t_i, c_j)$ which will be as close as possible to $f(t_i, c_j)$.

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 unknowing regularity. This sample is necessary for the classifier training and determining of its parameters values, 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 occurs.

In the set task each search request must response only to one class $c \in C$, and in this case the unambiguous classification will be possible.

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 decrease; 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 determination: the attributes set, their number and the methods of weight numbers calculating, and also determine the need of some algorithms parameters selection at 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.

2 Normalization of Search Query Data

Each search query text *T* consists of *S* that is the words array of the search query text and *W* that is the set of words without 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:

$$T = S \cup W = [word_1 \ldots word_n]. \tag{2}$$

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

For this goal, first to split up the search query for words or phrases "tokens" (perform "tokenization") is needed, taking into account the search query text specifics, i.e. phrase "technological process" should be perceived as one or two "tokens". To implement "tokenization" the N-gram is used [2-3]. The most common search queries when using tokenization are unigrams and bigrams. A comparative analysis of N-grams was conducted, and the results are given in Table 1.

Table 1. Results of search queries tokenization by N-grams

Search query	Unigrams	Bigrams	3-character N-grams
technological operation 3	["technological", "operation", "3"]	["technological operation", "3"]	["tec", "hno", "log", "ica", " l o", "per", "ati", "on ", " 3"]

The best variant would be using either unigrams or bigrams, because the symbol N-gram divides the search query text into unrelated letters that is difficult for further processing.

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 [5]. To do this we determine the weighing coefficients for each of the parameters. This can be performing using the methods of ranking and assigning points [5]. Expertise is conducted by the experts group of 20 people who are experts in this field. Experts set points by criteria and based on that information the matrix of points is formed. By the substitution of the matrix, weight factors for each of the parameters were calculated.

As a generalized estimate of the search query text the next expression can be used:

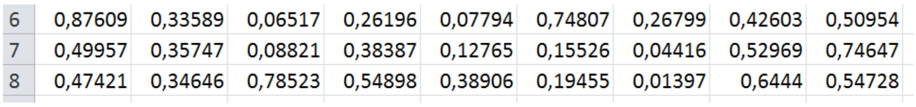
$$\lambda = 0.305 \cdot E + 0.18 \cdot O + 0.305 \cdot V + 0.21 \cdot P. \tag{3}$$

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.

After search query converting into the words sequence, converting them into the attributes vector can be started. 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. The weighing coefficients of search queries are standardized by taking into account value of *E, O, V* and *P*:

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

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 resulting dictionary matrix is stored as *Dataset.xls* file, and its fragment is shown in Fig. 1.



6	0,87609	0,33589	0,06517	0,26196	0,07794	0,74807	0,26799	0,42603	0,50954
7	0,49957	0,35747	0,08821	0,38387	0,12765	0,15526	0,04416	0,52969	0,74647
8	0,47421	0,34646	0,78523	0,54898	0,38906	0,19455	0,01397	0,6444	0,54728

Fig. 2. A fragment of the dictionary matrix for categorizing search queries

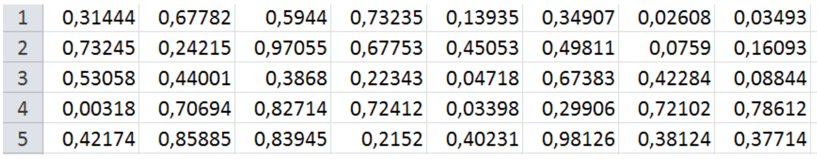
3 Training and Testing a Search Queries Classifier using Machine Learning

The developments of a classifier for the search queries classification are carried out using neural network learning. One of the types of neural networks is learned networks. 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.

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.

The neuron input layer is the search query text converted in the dictionary form, which is considered in section 2, i.e. it is a data matrix of 121×1119 elements (Fig. 1). Each neuron of the input layer is fed as a normalized number (from 0 to 1).

The neuron output layer is the search query characteristics (5 search queries categories in section 1), i.e. it is a categories vector of 5×1119 elements (Fig. 2) wich pre-processed similarly as dictionary.



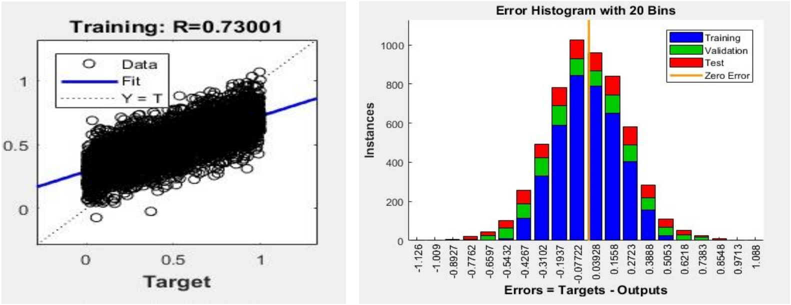
1	0,31444	0,67782	0,5944	0,73235	0,13935	0,34907	0,02608	0,03493
2	0,73245	0,24215	0,97055	0,67753	0,45053	0,49811	0,0759	0,16093
3	0,53058	0,44001	0,3868	0,22343	0,04718	0,67383	0,42284	0,08844
4	0,00318	0,70694	0,82714	0,72412	0,03398	0,29906	0,72102	0,78612
5	0,42174	0,85885	0,83945	0,2152	0,40231	0,98126	0,38124	0,37714

Fig. 2. A matrix fragment with neurons values for the output layer

After inputting 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 was changed.

Experimentally determined that the best result is given by a neuron network consisting of two layers. The hidden layer has a dimension of 30 neurons (that approximately is 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. 3a). 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.



a) regression evaluation b) histogram of errors

Fig. 3. Neural network learning outcomes assessment

As can be seen from Fig. 3a, the adjustment is good enough for all datasets, and the values of *R* in each case are 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.

We further evaluate the network productivity using the error histogram (Fig. 3b). In Fig. 3b, the blue columns indicate the learning data (*Training*), the green columns show check data (*Validation*), and there are test data (*Testing*) in red columns.

The histogram shows overshoot that are data points, where fitting is much worse than for most data. Fig. 3b 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. 3a). The first point corresponds to a point with the target of 0.597942219 and is displayed at 0.306553232320604.

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.

4 Conclusion

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 the result the dictionary was obtained in the matrix form of 121×1119 elements.

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.

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 the solving 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. N. Guand, X. Wang, “Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education”. USA: IGI Global, 2012.
2. D. H. Fisher, “Knowledge acquisition via incremental conceptual clustering. Machine Learning”, 1987. pp. 139-172.
3. X. Zhang, J. Zhao and Y. Le Cun, “Character-level convolutional networks for text classification,” *Proc. Neural Inform. Proc. Systems Conf. (NIPS 2015)*. Available: https://arxiv.org/abs/1509.01626. [Accessed November 28, 2018].
4. S. Vijayarani, J. Ilamathi and M. Nithya, “Preprocessing Techniques for Text Mining,” *Int. J. of Computer Science & Communication Networks*, vol. 5(1), pp. 7-16, 2014.
5. G. Laboreiro, L. Sarmento, J. Teixeira and E. Oliveira, “Tokenizing micro-blogging messages using a text classification approach”, *Proc.of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, October 26-26, 2010, Toronto, ON, Canada.