Commercial Content Distribution System Based on Neural Network and Machine Learning

Andriy Demchuk[0000-0001-6942-9436][1], Bohdan Rusyn[0000-0001-8654-2270][2], Liubomyr Pohreliuk[0000-0003-1482-5532][3], Aleksandr Gozhyj[0000-0002-3517-580X][4], Irina Kalinina[0000-0001-8359-2045][5], Lyubomyr Chyrun[0000-0002-9448-1751][6], Nataliya Antonyuk[0000-0002-6297-0737][7]

[1]Lviv Polytechnic National University, Lviv, Ukraine
[2-3]Karpenko Physico-Mechanical Institute of the NAS Ukraine
[4-5]Petro Mohila Black Sea National University, Nikolaev, Ukraine
[6-7]Ivan Franko National University of Lviv, Lviv, Ukraine
[7]University of Opole, Opole, Poland

Andrii.B.Demchuk@lpnu.ua[1], rusyn@ipm.lviv.ua[2], liubomyr@inoxoft.com[3], alex.gozhyj@gmail.com[4], irina.kalinina1612@gmail.com[5]

Abstract. In this paper, we consider the problem of designing an information system for methods and means of commercial distribution of information products, using a personalized approach to visitors based on categories and tags for interesting visitors of information products. The designed system is the methods and means of reorganization in the online store, with the core of the automatic recommendation of tags (categories) in the form of a neural network with controlled learning that provides the intelligence of the system as a whole. Providing a convenient site is key because online stores can help customers find the things they are looking for in a more versatile way. This allows visitors to manage their own buying experience, which helps to increase customer loyalty and makes them more inclined to return to the site for more purchases, which in turn greatly facilitates trade. The technologies of artificial intelligence will provide customers with better services and individual impressions. They also maximize the marketing efforts of the company, minimizing the need to spend money on ineffective advertising campaigns. The purpose of the intellectual system of Internet commerce is to provide unique content based on the approach of personalization and the use of tags.

Keywords: Information Resource, Information Products, SEO, Information Technology, Text Monitoring, Information Personalization, Information Products Distribution, Neural Network, Machine Learning, Sitecore CMS

1 Introduction

The main idea of the neural network is to simulate or copy in a simplified, but reliable way, the many densely integrated brain cells inside the computer to learn things, recognize patterns and make decisions in a human way [1]. An amazing thing about the neural network is that you do not have to program it; it will know everything by itself, just like the brain [2]. However, this is not a brain. Artificial neural networks are a type of artificial intelligence that tries to reproduce the work of the human brain [3]. They can learn from experience, recognize patterns and predict trends, they can tell which tactics people have been exposed to in a marketing campaign, and what should be discarded and rethought [4]. It's important to note that neural networks are usually simulation software, they are created by programming very simple computers that work very traditionally with conventional transistors and consistently logically connected so that they behave as if they are built from billions of high interactions [5]. The cells that work in parallel. Nobody ever tried to build a computer by connecting transistors in a tightly parallel structure, just like a human brain [6]. In other words, the neural network is different from the human brain just as the computer model of the weather differs from real clouds, snowflakes or sunlight [7]. Computer simulations are simply assemblies of algebraic variables and mathematical equations that bind together, in other words, numbers are stored in fields whose values are constantly changing [8].

2 Structure of Neural Networks

A typical neural network has something from several tens to hundreds, thousands, or even 1,000,000 of artificial neurons, which are called units that are arranged in series of layers, each of which connects to a layer on both sides [1,9]. Some of them, known as input units, is intended to receive various forms of information from the outside world that the network will attempt to identify, recognize, or otherwise handle [10]. Other units sit on the opposite side of the network and signal how it corresponds to the information received, they are known as output units [11]. Between input devices and output blocks there are one or more layers of hidden blocks that together form the majority of the artificial brain [12]. Most neural networks are fully connected, which means that each hidden block and each output block are connected to each block in layers on both sides [13]. The connection between one unit and the other is represented by a figure that is called a weight, which can be positive (if one unit excites the other) or negative (if one unit depresses another). The higher the value, the more one unit affects the other [14]. This corresponds to how actual brain cells provoke each other through tiny spaces called synapses. The fully connected neural network consists of input units (red), hidden units (blue) and output units (yellow), with all units connected in layers on both sides [15-18]. The entrances are left to the left, activate the hidden blocks in the middle, and output the exit from the right edge. The value of the connection between any two blocks is gradually adjusted as the network learns (Fig. 1) [19]. Information flows through the neural network in two ways [20] When she learns or works in normal mode (after training), samples of information enter the network through the input units that cause the layers of hidden elements, and they, in turn, come in the output blocks [21]. This general design is called the promotion network.

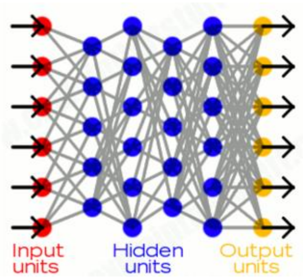

Fig. 1. View of the neural network

Not all items are running all the time. Each subsequent element receives its data from the neighboring element on the left and the number of nodes on which they are further transmitted multiplies the input data [22]. Each unit adds all the inputs it receives thus in the simplest type of network if the amount exceeds a certain threshold, the item is triggered and triggers items connected to the right side of it. In order for the neural network to learn, there must be an element of feedback - just as children learn, telling them what they are doing right or wrong [23]. In fact we are all user feedback. Think again, when you first learned to play the game like bowling. As you took a heavy bullet and roamed along the path, your brain watched as quickly as the ball moved, with which trajectory, and noticed how close you were to before falling into the pin [24-25]. Next time you mentioned that you did wrong, changed the movements accordingly, and tried to throw the ball a little better [26]. So you used the feedback to compare the results you wanted, with what actually happened, find out the difference between the two and used it to change what you did next time: "I need to throw more", "I need a little more left", "I need to release later", and so on. The greater the difference between the predicted and the actual result, the more radical you would change your actions [27]. Neural networks study things in exactly the same way, usually with the backup process called back-propagation [28]. This involves comparing the output data generated by the network with the data you need [29]. Using the difference between them to change the value of connections between units in the network, working from the output units through hidden units to the output units by going backwards [30]. Over time, reverse propagation causes the network to learn, reducing the difference between actual and predictable output until they exactly match, so the network reflects things exactly as it should be [31].

Once the network has been trained with a sufficient number of learning examples, it reaches a point where you can present it with a completely new set of input data you have never seen before and see how they react [32]. For example, suppose you have trained the network by showing it many chairs and tables of photographs presented in the proper way that it could understand, and say whether each one is a chair or a table [33]. By showing, say, 25 different chairs and 25 different tables, you draw a picture of some new design that you have not seen before - say, a chaise longue - and see what happens. Depending on how you have taught her, she will try to classify a new example or as a chair or table summarizing based on experience, just as in a person.

This does not mean that the neural network can simply "look" on the furniture and react immediately to them sensibly [34]. She does not behave like a man. The network actually does not look at the furniture. Enter the network, in essence, binary numbers: each input block or on or off. Therefore, if you have five input blocks, you can submit information about five different characteristics of different chairs using binary (yes / no) answers [35]. Questions can be [36]:

1. Do he have a back?
2. Does he have the upper hand?
3. Has a soft upholstery?
4. Can you sit on it comfortably for a long period?
5. Can you put many things on it?

A typical armchair will then be displayed as "Yes", "No", "Yes", "Yes", "No" or "10110" on the binary network, while the default table may be "No", "Yes", "No" "No", "Yes" or "0100". So, during the phase of learning the network simply looking at many numbers, such as 10110 and 01001 [37].

Based on this example, you may see many different neural network applications that recognize pattern recognition and make simple decisions about them. In aircraft, you can use a neural network as a base autopilot, with incoming devices read signals from various cockpit devices and output devices that modify the airplane control accordingly to safely keep the course. Inside the factory, you can use a neural network for quality control [38]. Suppose you do a laundry detergent where a complex chemical process takes place. You can measure the final detergent in different ways: its color, acidity, thickness or other, to supply these measurements to your neural network as input, and then the network decides whether to accept or reject the package.

There are many applications with neural networks in safety too. Let us say you work in a bank in which many thousands of credit card transactions pass through your computer system every minute. You need a quick automated way of detecting any operations that may be fraudulent and that is why the neural network is perfectly suited [39]. Your inputs would be such things as

1. Is the cardholder present really?
2. Have you used a valid PIN?
3. Does have five or more transactions, with this card in the last 10 minutes?
4. Is the card in another country from which it is registered used?

In the presence of sufficient prompts, the neural network can detect suspicious transactions, which allows the operator to staff more accurately examine them. In a very similar fashion, the bank can use the neural network to help it decide whether to give loans to people based on their past credit history, current income and seniority. In general, neural networks have made computer systems more useful, making them more human [40].

3 Neural network based on the Hashtable

Neural networks are formed from layers of similar neurons. Most neural networks have at least an input layer and output. The input layer represents the input pattern, and then the output pattern returns from the output layer. What happens between the input and output layers is a black box. At this stage, we are not yet interested in the internal structure of the neural network. There are many different architectures that determine what is going on between the input and output layers. You can design a neural network as something like a hash table in traditional programming. In traditional programming, the hash table is used to display the keys to values. Somewhat resembles a dictionary [41-43]. The following can be developed as a hash table:

```
"you hear" -> "perceived or perceived by the ear"
"run" -> "go faster than walk"
"write" -> "to form (letters or characters)
on the surface with a tool (like a pen) "
```

This is a reflection between the words and the definition of each word. This is a hash table, as you can see in any programming language. Used the key of the string, to another value of the string. You give the dictionary a key. It returns a value. That is how most neural networks operate. One neural network called two- way associative memory actually allows you to also pass the value and receive keys [44-49].

The programming hash tables use keys and values. For example, a template that is sent to the input layer of a neural network is very similar to the process of inputting a key into a hash table. Similarly, the value returned from the hash table as a template is similar to the one that returns from the source layer of the neural network. The comparison between the hash table and the neural network works well; however, the neural network is much larger than the hash table. What happens to the above hash table if you had a word that is not on the table? For example, if you had to pass in the key is "written". The hash table will return null or indicate that it is not possible to find the specified key. Neural networks do not return an empty result! They find the closest value. They will not only find the closest value, but they will also modify the conclusion to guess what would be in the absence of meaning. Therefore, if you are "input" to the above neural network, you will probably get what you would expect to "output". There is not enough data for the neural network to change the response since there are only three samples. This way, you will probably get one of the other keys on the output. Let us start with the fact that we consider the XOR operator as if it were a hash table. If you are not familiar with the XOR operator, it works the same way as operators AND and OR. For AND to be true, both sides must be true. In order for OR it to be true, any party must be true. In order for XOR to be true, both sides must be different from each other. The truth table for XOR is as follows [50-53]. To continue an example of hash tables, the truth table is presented as follows.

```
False XOR False = False
True XOR False = True
False XOR True = True
True XOR True = False
```

```
[0.0, 0.0] -> [0.0]
[1.0, 0, 0] -> [1.0]
[0.0, 1.0] -> [1.0]
[1.0, 1.0] -> [0.0]
```

These displays show the input and the ideal expected output for the neural network. Non-controlling training is also an iterative process. However, the calculation of the error is not so simple. You do not have an expected output, so you can not measure how uncontrolled the neural network is from the ideal exit, because you do not have an ideal exit. Often you will just repeat several iterations, and then try to use the network. If she needs additional training, then this training. Another very important aspect of the above training data is that it can be used in any order. The result "0" XOR "0" will be "0", regardless of how you look. This does not apply to all neural networks. For the XOR operator, we would probably have used the type of neural network, which is called feedforward [54-58].

4 Training Procedure and Designation

Tegger automatically assigns four false tags, two at the beginning and two at the end, for the target statement, and the neural network learns to automatically assign morphosyntactic descriptions that take into account the context, that is, two previously assigned tags and possible tags for the current and the next two words [59-64].

In our environment, a learning example consists of functions extracted for a single word within a state as an introduction, and this is a morphosyntactic description within this expression as a derivation. Functions are extracted from 5 words oriented to the current word. A vector that encodes or is assigned a morphosyntactic description, or its possible morphosyntactic descriptions characterizes one word. To encode possible morphosyntactic descriptions, use Equation 1, where each possible attribute has one corresponding position inside the encoded vector [65-67].

$$P(a \mid w) = \frac{C(w,a)}{C(w)}, \tag{1}$$

The following three vectors are used to encode possible morphosyntax descriptions for the current word and the following two words. During training, we also calculate the list of suffixes with the corresponding morphosyntax descriptions used during execution to create a possible morph-syntax vector for unknown words. When such words are found in the test data, we approximate their possible morphosyntax vector using the variant of the method proposed by Brants. When a tag is applied to a new statement, the system iteratively calculates the output morphosyntactic description for each individual word. After the tag is assigned to one word associated with that word, the vector changes so that it will have a value of 1 for each attribute present in its reassigned morphosyntax. As a result of coding of each attribute separately morphosyntax descriptions Tegher can assign new tags that have never been associated with the current word in the learning process. Although this is good behavior for working with unknown words, it often refers to the fact that it assigns an attribute value that is not valid for wordform. To overcome these types of errors, we use an additional list of words with their permitted morphosyntax descriptions [68-73]. For the word OOV, the list is calculated as an association of all morphosyntactic descriptions appearing with the suffixes that apply to that word [74-85]. When the tag has to assign an amorphous- syntactic description to a particular word, it chooses one of the morphosyntactic descriptions of possible word forms in its list with a word using the simple distance function:

$$\min_{e \in P} \sum_{k=0}^{n} |o_k - e_k| \qquad (2)$$

where: *P* is a list of all possible morphosyntax descriptions for a given word;
n is length of encoding of morphosyntactic descriptions (110 bits);
o is output of the neural network for the current word;
e is binary coding for morphosyntactic descriptions in *P*.

5 The Evolution of Artificial Intelligence in E-Commerce

Client-Oriented Search. Amir Königsberg is the current CEO of Twiggle, a business that allows e-commerce search engines to think like humans [1-3]. Consumers often refuse e-commerce, because the reflected products often do not meet the needs of the user [4-9]. To solve this problem, Twiggle uses natural language processing to improve search results for online shoppers. Another business that is trying to improve e-commerce search is Clarifai in the United States. Clarifai's early work focused on visual search elements, and their website says their software is "artificial intelligence with vision." They allow developers to create smarter applications that "see the world as you", giving companies the ability to develop client-centric experience through enhanced image and video recognition [1-6]. By using machine-learning technology, software with artificial intelligence automatically adds tags, organizes and searches content by marking the properties of an image or video. The use of artificial intelligence gives companies a competitive advantage and is available to developers or businesses of any size or budget. An excellent example is the recent update of Pinterest to its Chrome extension, which allows users to select an object on any photo on the Internet, and then use Pinterest to search for similar objects using the image recognition software. Not only does Pinterest present a new search experience with artificial intelligence. New software platforms that control e-commerce create innovative visual search capabilities. As well as finding relevant products, artificial intelligence allows buyers to find additional products, whether they are size, color, shape, fabric or even brand. The visual capabilities of such software are outstanding really. Initially getting visual signals from downloaded images, the software can successfully help the customer find the product. All that is visual, thanks to artificial intelligence, consumers can easily find such things through e-commerce stores [1-13].

Personal Contact with Chatbots. Trade now focuses on people, not on the mass market. For consumers there are plenty of places, where you can make purchases on the Internet. Many e-commerce retailers are sophisticated becoming more with their capabilities of artificial intelligence to attract attention, and one widely developed approach is called "spoken trades". In the e-commerce world, it combines visual, vocal, written, and intellectual capabilities. The needs of consumers are developing rapidly, in turn; retailers try not to lag behind. If brands want to survive, this is one of the priority business strategies that need to be met. The use of artificial intelligence with the help of "chat-bot" is just one of the ways to spoken [1-13].

Virtual Assistants. Sometimes we need help online. We are all familiar with Siri, Google Now and Alexa, and they have successfully introduced us to the idea of talking with a phone, a laptop. Virtual assistants relate to the processing of natural language and the ability of the machine to interpret what people are saying in words or text. So, what does this mean for e-commerce retailers? Let's take a look at the virtual assistant Amazon, Alexa. Their virtual assistant, who has recently become one of the most prominent voters in commerce, has been successfully integrated into their own Amazon products as well as products from other manufacturers. For example, using Alexa on Echo Amazon, customers can open local concerts for the next weekend through StubHub, order a taxi through Uber, or even order delivery of dinner with Domino and track the status of the order in real time. Even more popular 1-800-Flowers in the US even allow consumers to send flowers to their loved ones with a voice. Virtual assistants influence how customers buy and provide a creative opportunity for e-commerce retailers to take advantage of these benefits [1-13].

Recommendations for clients. By using artificial intelligence, brands can scan more intelligently and efficiently data to predict customer behavior, and offer relevant and useful recommendations to individual consumers [1-13]. This level of intelligence is vital to meeting the needs of consumer purchases. Starbucks is actively engaged in this process, using artificial intelligence to analyze all the data that it has gathered for its customers, and providing more personalized offers. For example, recently launched My Starbucks Barista, which uses artificial intelligence to allow customers to place orders using voice commands or messages. The algorithm uses a variety of inputs, including account information, consumer preferences, purchase history, third-party data and context information. This allows coffee houses to create and provide more personalized messages and recommendations to their customers. The desire of many e-commerce companies is to provide the best way to online distribution on the Internet, offering customers an impeccable way to find the products they are actively seeking. The recommendation process is widely practiced by retailers of e-commerce to help customers find the best solution. For example, Amazon makes recommendations to users depending on their activity on the site and any previous purchases. Netflix makes recommendations for television and cinema based on user interaction with categories such as drama, comedy, and more. Providing tailor-made advice helps people find what they are looking.

6 System Analysis and Reasoning of the Problem

The growth of the Internet, the amount of information and products available on individual websites, has increased exponentially (Fig. 2).

Today, Google indexes over 30 trillion web pages, Amazon sells 232 million products, Netflix has over 10,000 titles, and more than 100 hours of video uploaded to YouTube on a weekly basis. Although such large assortments provide consumers with a sufficient choice, they also complicate the placement of the exact product or information they are looking for.

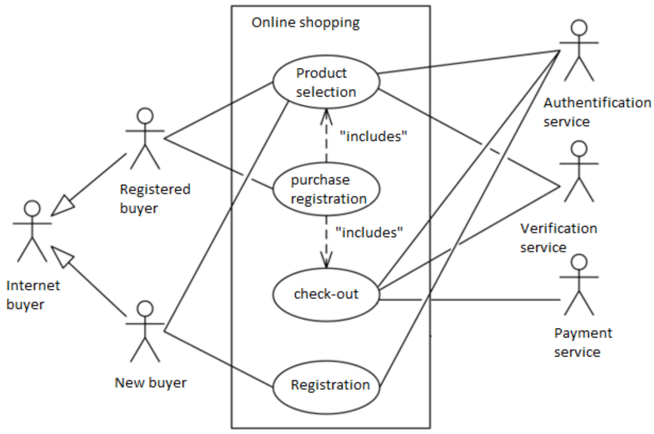

Fig. 2. UML Diagram for Online Store

Therefore, most businesses have adopted a personalization model based on a query that helps consumers find the product or information that best suits their needs. To address the tasks described above, an e-commerce website needs to develop an intelligent decision-making system that, when adding content to the reader of each new product in the e-commerce application provides an expanded product description and will provide the appropriate recommended categories and tags using synonymous variations. The synonym row will be determined by means of a neural network and a "list of algorithms". An example of application will be implemented based on. Net CMS Sitecore. Which owns the personalization tools available to expand and add their own development based on the available core- functionality (Fig. 3-6). Personalization is a method of displaying targeted, relevant content for users based on their characteristics and behavior, such as location, gender, or previous visits. With personalization, you can make sure that the right content reaches the right users, for example by showing, hiding or configuring content.

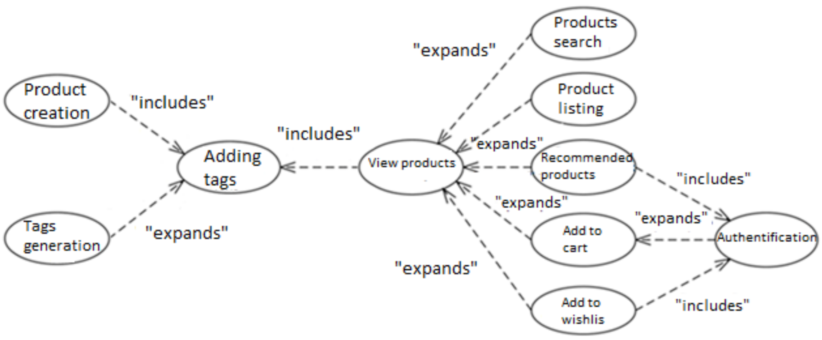

Fig. 3. Using an online store

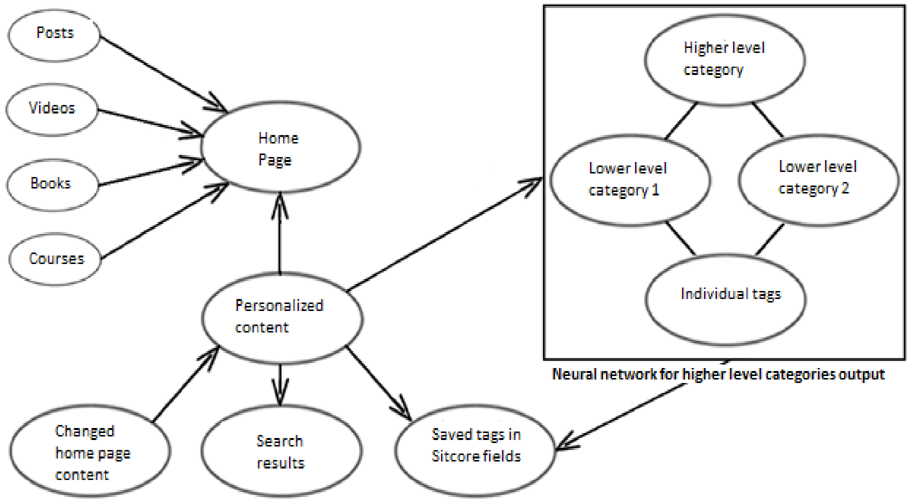

Fig. 4. Personalize content

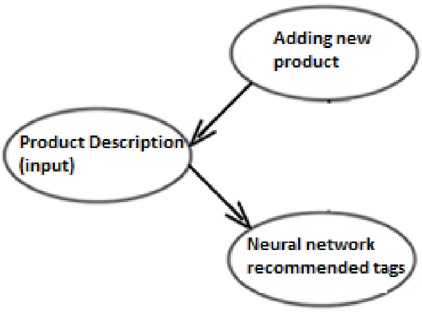

Fig. 5. Adding a product

Among other things, you can use personalization to:

- Show other content for users based on their geographic location.
- Hide user registration form that has previously filled out the form.
- Edit text on a banner website based on a user's site link.
- Controlled Training Neighborhood and Breakdowns on Tokens

Text tokening is a process of splitting a string containing text into separate tokens. As a result, there is a reduction in the number of words to the abbreviated root of the word, which makes it easy to compare the equality of similar words. Tagging is the definition of which part of each word is in the input text. Labelling is complicated by many words that have different parts of the speech depending on the context (for example, bank the airplane, the river Bank, etc.). The code in this section can be found in the src / com / knowledge books / NLP / fast tag / FastTag.java ZIP and src / com / knowledge books / NLP / util / Tokenizer.java ZIP files. The required data files are located in the lexicon.txt file (for processing English text) and lexicon medpost.txt (for medical processing text). Before processing any text, we need to split the text into separate tokens. Tokens can be words, numbers, and punctuation symbols. The Tokenizer class has two static methods; both take the input string for a token and return a list of tokens. The second method has an additional argument for determining the maximum number of tokens:

```
static public List<String> wordsToList(String s)

static public List<String> wordsToList(String s, int maxR)
```

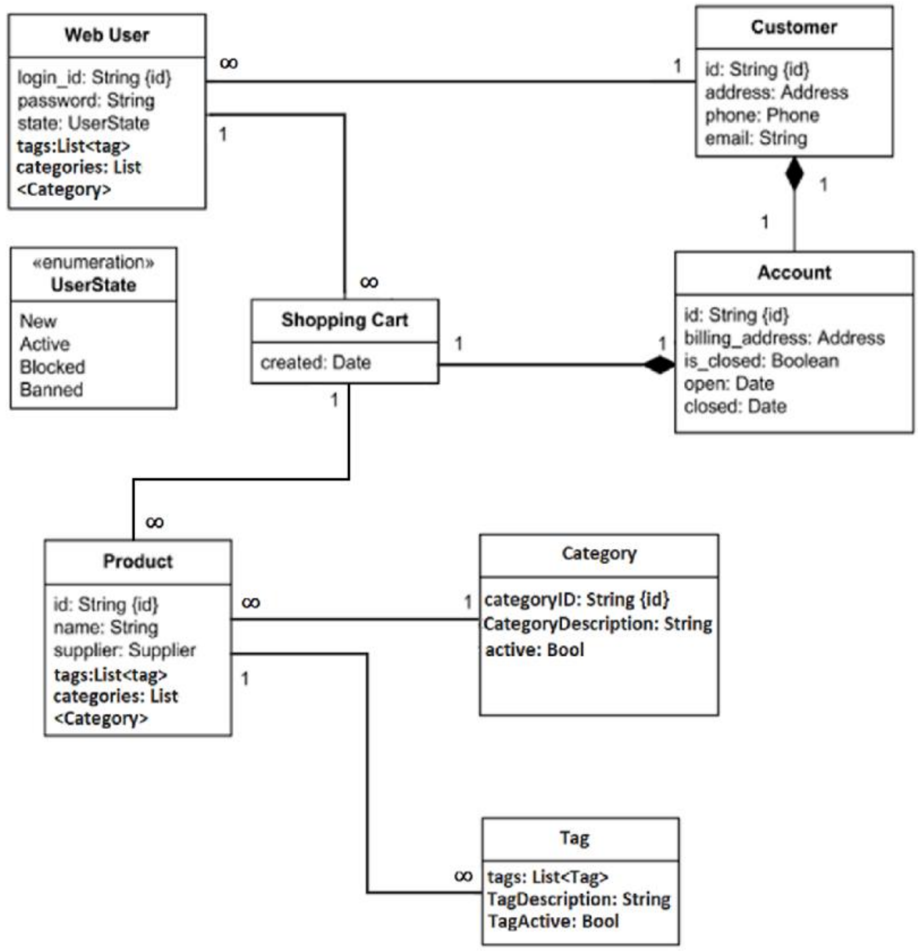

Fig. 6. ER Chart for Online Store.

The src/com/knowledgebooks/nlp/fasttag/README.txt file contains information on where to get the original Eric Brillan tagging system, as well as tag definitions for both of its Lexicon and Medpost vocabulary. Examples of tag description:

Tag	Description	Examples
NN	singular noun	dog
NNS	plural noun	dogs
NNP	singular proper noun	California
NNPS	plural proper noun	Watsons
CC	conjunction	and, but, or
CD	cardinal number	one, two
DT	determiner	the, some
IN	preposition	of, in, by
JJ	adjective	large, small, green
JJR	comparative adjective	bigger
JJS	superlative adjective	biggest
PP	proper pronoun	I, he, you
RB	adverb	slowly
RBR	comparative adverb	slowest
RP	particle	up, off
VB	verb	eat
VBN	past participle verb	eaten
VBG	gerund verb	eating
VBZ	present verb	eats
WP	wh* pronoun	who, what
WDT	wh* determiner	which, that

Fig. 7. The most commonly used tags

The following list shows a sample code snippet using this class with the output:

```
String text = "The ball, rolling quickly, went down the hill.";
List<String> tokens = Tokenizer.wordsToList(text);
System.out.println(text);
for (String token : tokens)
      System.out.print("\""+token+"\" ");
System.out.println();
```

This code snippet displays the following:

```
The ball, rolling quickly, went down the hill.
"The" "ball" "," "rolling" "quickly" "," "went" "down" "the" "hill" "."
```

For many applications, it's best to pull out word tokens to simplify the comparison of similar words. For example, "run" and "launch" everything to "run". The set of words we will use is in the src / public file domain / stemmer.java. At the end of the class, there are two convenient APIs, one to create a string of several words, and one to create one token of the words:

```
public List<String> stemString(String str)
public String stemOneWord(String word)
```

FastTag used machine-learning techniques to learn the rules of transition for text tags using handwritten text as an example of learning. The Java version is located in the file src / com / knowledge books / nlp / fasttag /FastTag.java.
So, we will process the string "fair JJ NN RB" as a hash key "fair", and the hash value is an array of rows (currently only the first value is used, but I save other values for

future use): `["JJ", "NN", "RB"]`. The list the remaining eight rules in the short syntax here:

```
One: the index i is the cycle variable in the range [zero, the number of
tokens of the word is one], and the word is the current word in index i:
// rule 1: DT, {VBD | VBP} --> DT, NN
if (i > 0 && ret.get(i - 1).equals("DT")) {
    if (word.equals("VBD") || word.equals("VBP") || word.equals("VB")) {
        ret.set(i, "NN");
    }
}
Two: convert a noun to the number (CD) if «.» appears in the word.
Three: convert a noun to a valid timestamp in the past, if words.get (i)
ends with «ed».
Four: convert any type into an adverb, if it ends with «ly».
Five: convert the noun (NN or NNS) to the adjective, if it ends with «al».
Six: convert a noun to a verb if the preceding word is «would».
Seven: if the word has been classified as a total, and ends with the
symbol «s», the type is set to the plural common noun (NNS).
Eight: converting a noun into an active verb (for example, gerund).
```

This rule states are accompanied that if the determinant (DT) in the word token of the index i-1 by the last time the verb (VBD) or the current verb (VBP), then the type of the tag changes to i for "NN".

7 Conclusions

In this paper, we consider the problem of designing an information system for methods and means of commercial distribution of information products, using a personalized approach to visitors based on categories and tags for interesting visitors of information products. The designed system is the methods and means of reorganization in the online store, with the core of the automatic recommendation of tags (categories) in the form of a neural network with controlled learning that provides the intelligence of the system as a whole. The developed system has classes and subclasses to which real information products will belong, logical links between them are built with which intellectual feed of the content should take place. The system of commercial distribution of information products in the future will be able to bring real income to its owner, which will be in demand among users of the World Wide Web. It should also be noted that the topic of Internet commerce in the context of e-business is more than ever relevant in our time, the time of rapid development of information technology, as to me the future of commerce on the Internet. It is already very popular to order any copyrighted information products. Therefore, who will understand this trend in the market of commerce in general, and will successfully be able to fit into it - will receive serious dividends.

References

1. Sitecore Documentation: Access all the latest Sitecore documentation. Available at: https://doc.sitecore.com

2. Demchuk, A., Lytvyn, V., Vysotska, V., Dilai, M.: Methods and Means of Web Content Personalization for Commercial Information Products Distribution. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 332–347. (2020)
3. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V., Kovalchuk, R., Petruchenko, O., Dzyubyk, L., Sokulska, N.: Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. In: Eastern-European Journal of Enterprise Technologies, 2(2-98), 15-34. (2019)
4. Mobasher, B.: Data mining for web personalization. In: The adaptive web, 90-135. (2007)
5. Dinucă, C., Ciobanu, D.: Web Content Mining. In: Economics, 85. (2012).
6. Xu, G., Zhang,Y., Li, L.: Web content mining. Springer, 71-87. (2011)
7. Khribi, M. K., Jemni, M., Nasraoui, O.: Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In: International Conference on Advanced Learning Technologies, 241-245. (2008).
8. Ferretti, S., Mirri, S., Prandi, C., Salomoni, P.: Automatic web content personalization through reinforcement learning. In: Journal of Systems and Software, 121, 157-169 (2016)
9. Lavie, T., Sela, M., Oppenheim, I., Inbar, O., Meyer, J.: User attitudes towards news content personalization. In: Int. journal of human-computer studies, 68(8), 483-495. (2010).
10. Fredrikson, M., Livshits, B. Repriv: Re-imagining content personalization and in-browser privacy. In: Symposium on Security and Privacy, 131-146. (2011).
11. Chang, C. C., Chen, P. L., Chiu, F. R., Chen, Y. K.: Application of neural networks and Kano’s method to content recommendation in web personalization. In: Expert Systems with Applications, 36(3), 5310-5316. (2009).
12. Galushka, I., Shcherbak, S.: Devising a mathematical model for pattern-based enterprise data integration. Eastern-European Journal of Enterprise Technologies, 2(9), 59-64 (2015)
13. Shcherbak, S.S.: Interoperability web application models based on microformats. In: 21st International Crimean Conference: Microwave and Telecommunication Technology, Conference Proceedings, 57-58 (2011)
14. Mirri, S., Prandi, C., Salomoni, P.: Experiential adaptation to provide user-centered web content personalization. In: Proc. IARIA Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC2013), 31-36. (2013).
15. Fernandez-Luque, L., Karlsen, R., Bonander, J.: Review of extracting information from the Social Web for health personalization. In: Journal of medical Internet research, e15 (2011).
16. Ho, S. Y., Bodoff, D., Tam, K. Y.: Timing of adaptive web personalization and its effects on online consumer behavior. In: Information Systems Research, 22(3), 660-679. (2011).
17. Uchyigit, G., Ma, M. Y.: Personalization techniques and recommender systems. In: World Scientific, Vol. 70, (2008).
18. Zhang, H., Song, Y., Song, H. T.: Construction of ontology-based user model for web personalization. In: Int. Conf. on User Modeling, Springer, Berlin, Heidelberg, 67-76. (2007).
19. Mehtaa, P., Parekh, B., Modi, K., Solanki, P.: Web personalization using web mining: concept and research issue. In: International Journal of Information and Education Technology, 2(5), 510. (2012).
20. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136 (2018)
21. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)

22. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216 (2018)
23. Korobchinsky, M., Vysotska, V., Chyrun, L., Chyrun, L.: Peculiarities of Content Forming and Analysis in Internet Newspaper Covering Music News, In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 52-57 (2017).
24. Naum, O., Chyrun, L., Kanishcheva, O., Vysotska, V.: Intellectual System Design for Content Formation. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 131-138 (2017)
25. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city. In: CEUR Workshop Proceedings (Computational linguistics and intelligent systems), 2255, 239-255. (2018).
26. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: 2015 Xth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), 120-123. (2015).
27. Lytvyn, V., Vysotska, V.: Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies, CSIT'2015, 115-119 (2015)
28. Kravets, P.: The control agent with fuzzy logic. In: Perspective Technologies and Methods in MEMS Design, MEMSTECH'2010, 40-41 (2010)
29. Vysotska, V.: Linguistic Analysis of Textual Commercial Content for Information Resources Processing. In: Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET'2016, 709–713 (2016)
30. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 2, 113-116 (2018)
31. Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., Burov, Y.: Information resources processing using linguistic analysis of textual content. In: Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications, Romania, 573-578, (2017)
32. Vysotska, V., Chyrun, L., Chyrun, L.: Information Technology of Processing Information Resources in Electronic Content Commerce Systems. In: Computer Science and Information Technologies, CSIT'2016, 212-222 (2016)
33. Vysotska, V., Rishnyak, I., Chyrun L.: Analysis and evaluation of risks in electronic commerce, CAD Systems in Microelectronics, 9th International Conference, 332-333 (2007).
34. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)
35. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 124-128 (2015)
36. Vysotska, V., Chyrun, L., Chyrun, L.: The Commercial Content Digest Formation and Distributional Process. In: Computer Science and Information Technologies, Proc. of the XI–th Int. Conf. CSIT'2016, 186-189 (2016)
37. Vasyl, Lytvyn, Victoria, Vysotska, Dmytro, Dosyn, Roman, Holoschuk, Zoriana, Rybchak: Application of Sentence Parsing for Determining Keywords in Ukrainian Texts. In: Computer Science and Information Technologies, CSIT, 326-331 (2017)
38. Rusyn, B., Lytvyn, V., Vysotska, V., Emmerich, M., Pohreliuk, L.: The Virtual Library System Design and Development, Advances in Intelligent Systems and Computing, 871, 328-349 (2019)

39. Rusyn, B., Vysotska, V., Pohreliuk, L.: Model and architecture for virtual library information system, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018, 37-41 (2018)
40. Burov, Y., Vysotska, V., Kravets, P. Ontological approach to plot analysis and modeling. CEUR Workshop Proceedings, Vol-2362, 22-31 (2019)
41. Zdebskyi, P., Vysotska, V., Peleshchak, R., Peleshchak, I., Demchuk, A., Krylyshyn, M.: An Application Development for Recognizing of View in Order to Control the Mouse Pointer. In: CEUR Workshop Proceedings, Vol-2386, 55-74. (2019)
42. Lytvyn, V., Kuchkovskiy, V., Vysotska, V., Markiv, O., Pabyrivskyy, V. Architecture of System for Content Integration and Formation Based on Cryptographic Consumer Needs. In: Computer Sciences and Information Technologies (CSIT). (2018).
43. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Senyk, A. et. al.: Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian. In: Eastern-European Journal of Enterprise Technologies, 6 (2 (96)), 19–31. (2018).
44. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The Method of Web-Resources Management Under Conditions of Uncertainty Based on Fuzzy Logic. In: Conference on Computer Sciences and Information Technologies (CSIT). (2018).
45. Lytvyn, V., Vysotska, V., Uhryn, D., Hrendus, M., Naum, O.: Analysis of statistical methods for stable combinations determination of keywords identification. In: Eastern-European Journal of Enterprise Technologies, 2 (2 (92)), 23–37. (2018).
46. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Kovalchuk, R., Huzyk, N.: Development of the linguometric method for automatic identification of the author of text content based on statistical analysis of language diversity coefficients. Eastern-European Journal of Enterprise Technologies, 5 (2 (95)), 16–28. (2018).
47. Lytvyn, V., Vysotska, V., Pukach, P., Vovk, M., Ugryn, D.: Method of functioning of intelligent agents, designed to solve action planning problems based on ontological approach. In: Eastern-European Journal of Enterprise Technologies, 3 (2(87)), 11–17. (2017).
48. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The risk management modelling in multi project environment. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). (2017).
49. Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., Rishnyak, I.: The Contextual Search Method Based on Domain Thesaurus. In: Advances in Intelligent Systems and Computing II, 310–319. (2018).
50. Veres, O., Rusyn, B., Sachenko, A., Rishnyak, I.: Choosing the method of finding similar images in the reverse search system. In: CEUR Workshop Proceedings (Computational linguistics and intelligent systems), 2136, 99–107. (2018).
51. Lytvyn, V., Vysotska, V., Pukach, P., Bobyk, I., Pakholok, B.: A method for constructing recruitment rules based on the analysis of a specialist's competences. In: Eastern-European Journal of Enterprise Technologies, 6 (2 (84)), 4–14. (2016).
52. Chyrun, L., Vysotska, V., Kis, I., Chyrun, L.: Content Analysis Method for Cut Formation of Human Psychological State. In: Data Stream Mining & Processing (DSMP). (2018).
53. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies. In: Advances in Intelligent Systems and Computing, 206–221. (2019).
54. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: Content linguistic analysis methods for textual documents classification. In: International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT). (2016).

55. Lytvyn, V., Vysotska, V., Kuchkovskiy, V., Bobyk, I., Malanchuk, O., Ryshkovets, Y. et. al.: Development of the system to integrate and generate content considering the cryptocurrent needs of users. In: Eastern-European Journal of Enterprise Technologies, 1 (2 (97)), 18–39. (2019).
56. Vasyl, L., Victoria, V., Dmytro, D., Roman, H., Zoriana, R.: Application of sentence parsing for determining keywords in Ukrainian texts. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). (2017).
57. Mukalov, P., Zelinskyi, O., Levkovych, R., Tarnavskyi, P., Pylyp, A., Shakhovska, N.: Development of System for Auto-Tagging Articles, Based on Neural Network. In: CEUR Workshop Proceedings, Vol-2362, 106-115. (2019)
58. Shakhovska, N. B., Noha, R. Y.: Methods and tools for text analysis of publications to study the functioning of scientific schools. In: Journal of Automation and Information Sciences, 47(12). (2015)
59. Shakhovska, N., Shvorob, I.: The method for detecting plagiarism in a collection of documents," Computer Sciences and Information Technologies (CSIT), 142-145. (2015)
60. Arzubov, M., Shakhovska, N., Lipinski, P.: Analyzing ways of building user profile based on web surf history. In: Computer Sciences and Information Technologies (CSIT), 1, 377-380. (2017)
61. Shakhovska, N., Basystiuk, O., Shakhovska, K.: Development of the Speech-to-Text Chatbot Interface Based on Google API. In: CEUR Workshop Proceedings, Vol-2386, 212-221. (2019)
62. Bobalo, Y., Stakhiv, P., Mandziy, B., Shakhovska, N., Holoschuk, R.: The concept of electronic textbook "Fundamentals of theory of electronic circuits. In: Przegląd Elektrotechniczny, 88 NR 3a/2012, 16-18. (2012)
63. Mashkov V.A., Barabash O. V.: Self-checking and self-diagnosis of module systems on the principle of walking diagnosis kernel. In: Engineering Simulation, 15, 43-51. (1998)
64. Kravets, P.: The Game Method for Orthonormal Systems Construction. In: The Experience of Designing and Applications of CAD Systems in Microelectronics. (2007).
65. Kravets, P.: Game Model of Dragonfly Animat Self-Learning. Perspective Technologies and Methods in MEMS Design, 195–201. (2016).
66. Lytvyn, V., Sharonova, N., Hamon, T., Cherednichenko, O., Grabar, N., Kowalska-Styczen, A., Vysotska, V.: Preface: Computational Linguistics and Intelligent Systems (COLINS-2019). In: CEUR Workshop Proceedings, Vol-2362. (2019)
67. Emmerich, M., Lytvyn, V., Yevseyeva, I., Fernandes, V. B., Dosyn, D., Vysotska, V.: Preface: Modern Machine Learning Technologies and Data Science (MoMLeT&DS-2019). In: CEUR Workshop Proceedings, Vol-2386. (2019)
68. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O.: Automated Monitoring of Changes in Web Resources. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 348–363. (2020)
69. Lytvyn, V., Vysotska, V., Rzheuskyi, A.: Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. In: CEUR Workshop Proceedings, Vol-2392, 147-171. (2019)
70. Babichev, S., Taif, M.A., Lytvynenko, V., Osypenko, V.: Criterial analysis of gene expression sequences to create the objective clustering inductive technology. In: 2017 IEEE 37th International Conference on Electronics and Nanotechnology, ELNANO 2017 - Proceedings, art. no. 7939756, pp. 244-248. (2017)

71. Babichev, S.A., Gozhyj, A., Kornelyuk, A.I., Lytvynenko, V.I. Objective clustering inductive technology of gene expression profiles based on SOTA clustering algorithm. In: Biopolymers and Cell, 33 (5), pp. 379-392. (2017)
72. Lytvynenko, V., Wojcik, W., Fefelov, A., Lurie, I., Savina, N., Voronenko, M. et al.: Hybrid Methods of GMDH-Neural Networks Synthesis and Training for Solving Problems of Time Series Forecasting. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 513-531. (2020)
73. Rzheuskyi, A., Gozhyj, A., Stefanchuk, A., Oborska, O., Chyrun, L., Lozynska, O., Mykich, K., Basyuk, T.: Development of Mobile Application for Choreographic Productions Creation and Visualization. In: CEUR Workshop Proceedings, 2386, 340-358. (2019)
74. Rzheuskyi, A., Kunanets, N., Kut, V.: Methodology of research the library information services: The case of USA university libraries. Advances in Intelligent Systems and Computing, 689, 450-460 (2018)
75. Rzheuskyi, A., Kunanets, N., Kut, V.: The analysis of the United States of America universities library information services with benchmarking and pairwise comparisons methods. In: Computer Sciences and Information Technologies, 417-420 (2017).
76. Rzheuskyi, A., Matsuik, H., Veretennikova, N., Vaskiv, R.: Selective Dissemination of Information – Technology of Information Support of Scientific Research. Advances in Intelligent Systems and Computingб 871, 235-245 (2019)
77. Rzheuskiy, A., Veretennikova, N., Kunanets, N., Kut, V.: The information support of virtual research teams by means of cloud managers. International Journal of Intelligent Systems and Applications, 10(2), 37-46 (2018)
78. Vysotska, V., Chyrun, L.: Methods of information resources processing in electronic content commerce systems. In: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
79. Andrunyk, V., Chyrun, L., Vysotska, V.: Electronic content commerce system development. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
80. Alieksieieva, K., Berko, A., Vysotska, V.: Technology of commercial web-resource processing. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
81. Chyrun L., Leshchynskyy E., Lytvyn V., Rzheuskyi A.,Vysotska V., Borzov Y.: Intellectual Analysis of Making Decisions Tree in Information Systems of Screening Observation for Immunological Patients. In: CEUR Workshop Proceedings, Vol-2362, 281-296. (2019)
82. Lytvyn, V., Vysotska, V., Shakhovska, N., Mykhailyshyn, V., Medykovskyy, M., Peleshchak, I., Basto Fernandes, V., Peleshchak, R., Shcherbak, S.: A Smart Home System Development. In: Advances in Intelligent Systems and Computing IV, 1080, 804-830. (2020)
83. Lytvyn V., Burov Y., Kravets P., Vysotska V., Demchuk A., Berko A., Ryshkovets Y., Shcherbak S., Naum O.: Methods and Models of Intellectual Processing of Texts for Building Ontologies of Software for Medical Terms Identification in Content Classification. In: CEUR Workshop Proceedings, Vol-2362, 354-368. (2019)
84. Oksanych, I., Shevchenko, I., Shcherbak, I., Shcherbak, S.: Development of specialized services for predicting the business activity indicators based on micro-service architecture. In: Eastern-European Journal of Enterprise Technologies, 2(2-86), 50-55 (2017)
85. Lytvyn, V., Kowalska-Styczen, A., Peleshko, D., Rak, T., Voloshyn, V., Rainer Noennig, J., Vysotska, V., Nykolyshyn, L., Pryshchepa, H.: Aviation Aircraft Planning System Project Development. In: Advances in Intelligent Systems and Computing IV, Springer, Cham, 1080, 315-348. (2020)