Development of Information System for Textual Content Categorizing Based on Ontology

Victoria Vysotska[0000-0001-6417-3689], Vasyl Lytvyn[0000-0002-9676-0180], Yevhen Burov[0000-0001-6124-3995], Pavlo Berezin[0000-0003-1869-5050], Michael Emmerich[0000-0002-7342-2090], Vitor Basto Fernandes[0000-0003-4269-5114]

[1-4]Lviv Polytechnic National University, Lviv, Ukraine
[2]Silesian University of Technology, Gliwice, Poland
[5]Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
[6]University Institute of Lisbon, Portugal
Victoria.A.Vysotska@lpnu.ua[1], Vasyl.V.Lytvyn@lpnu.ua[2], yevhen.v.burov@lpnu.ua[3], pavlo.berezin@gmail.com[4], emmerix@gmail.com[5],vitor.basto.fernandes@gmail.com[6]

Abstract. The methods and means of using ontologies within systems for the categorization of textual content were created. Also, a method for optimizing the definition of which rubrics best relate to a certain text content was developed. The intellectual system that will use the methods developed earlier, as well as other research results was implemented. The results will allow users to easily filter their text content. The system developed has an intuitive user interface.

Keywords: ontology, content, text categorization.

1 Introduction

The rapid development of the World Wide Web, in particular the increase in the amount of content, raises the difficult task of analyzing such amount of content [1]. First of all, it defines the information that a person will ultimately get. Precisely because of this, the task of automating the rubricating of text content is very important [2]. The main problem of manual rubricating is that it takes a lot of time and effort on a person who does it. Also, with manual heading, there is no unification of headings between people who read text. Automatic rubricating solves this problem, because it will allow [3]:

- To simplify the search for required information.
- Discard the need for manual rubrication.
- Unify rubrics.
- Improve understanding of the content itself.

The aim of the work is development of a system for categorizing text content. To address this goal, the following research tasks structure was developed:

1. Research of various methods of constructing a system of heading text content.
2. Exploration of various languages describing ontologies.
3. Investigation of the finished decisions of the heading of text content.
4. Search for ready ontologies.
5. Construction of the method of allocating the most relevant rubrics and categories among the system received.

The object of research is the process of creating intelligent systems for categorizing text content for convenient categorization of text records of a large size.

The subject of study is means and methods of creating intelligent systems for rubricating of text content based on the ontological approach for analysis and rubricating the text data arrays. This system should free the text's authors from categorizing their text, as well as simplify the search for content for a specific topic for users.

2 Analytical review of literature

2.1 The analysis of key concepts

Analyzing the basic methods of text content rubricating, one can conclude that methods using ontologies are among the best, even though these systems do not use all the features and benefits of ontologies [1-9]. Compared with other methods of constructing knowledge bases, the advantages of using ontologies are obvious, since it is ontologies that are the standard of knowledge engineering and have proven themselves as the best method for representing objective knowledge[10-16] . In particular, the following tasks are not solved: criteria for filling and optimizing ontologies; modeling processes for processing information resources and the emergence of new knowledge based ontologies; assessment of the novelty of ontology knowledge. It is the solution of these tasks that will allow to build effective application information systems for the processing of text content, namely, its rubricating [17-21].

Under the formal model of ontology *O*, three of the following form are understood: $O=<C,R,F>$, where *C* is finite set of concepts (concepts, terms) of software [1-9], which sets the ontology *O*; $R:C->C$ is finite set of relationships between concepts (concepts, terms) of a given software; *F* is finite set of functions interpretation (axiomatization, restrictions) defined on the ontology concepts or relations *O*. It should be noted that the set *C* is limited to finiteness and non-emptiness, and *F* and *R* must be also finite. One of the most common ways to represent an ontology is the graph (conceptual graph). In this graph, the vertices are software concepts, arcs are the relation between concepts. The vertices can be interpreted or not interpreted. Interpretation depends on whether the axioms of concepts are defined [22-26].

The arcs are divided into vertical and horizontal. Vertical relationships are given by taxonomy of software concepts. To set a horizontal relationship, you need to define the set of values and the area of determination of the relation [27-31].

The structure of ontology, in its general form, is a set of four categories of elements: concept, relation, axioms and individual instances [32-36].

Concepts (classes) are general categories that are hierarchically arranged. Classes can be regarded as the conceptualization of all representatives of a certain entity. Each class describes a group of instances that have certain common properties. Concepts are combined with each other through different relationships. The most common, among systems using ontologies, the relationship type is categorization, that is, the allocation of a certain class under the appropriate category [37-42].

Instances - are other elements of ontology, which are representatives of a certain class, that is, the elements that belong to a particular category. Elements of ontology have a specific hierarchy. Instances, like specific representatives, occupy the lower level of the hierarchy. Above them are categories (concepts). Above categories there are relations between them. The rules and axioms combine all these levels of the hierarchy. To construct an ontological model, first of all, it is necessary to determine the hierarchy of concepts (set *C*). When constructing an infological model in the form of an ontology, experts should be involved in subject areas, which as a rule will use abstraction and combination. The main method for constructing ontologies is the classification. Classification is a method for organizing knowledge. In object-oriented analysis, the simplicity of the architecture of the infological model of the system is achieved by identifying the general properties of objects, through which represent the key abstractions and mechanisms. However, there are no strict classification methods and well-organized rules now on how to classify objects and their classes. Therefore, there are no such concepts as "ideal class hierarchy", "proper classification of objects", since the choice of classes is a compromise solution. It is through the classification that we combine different objects into one group, according to their structure or behavior. It should be noted that no system of rubrication of text content can be achieved without parsing and identifying keywords. Keyword is a word, or a constant expression of a natural language, used to express some aspect of a document's content (or query); a word that has a significant semantic load. It can serve as a key when searching for information. Parsing (parsing) is a process of analyzing the input sequence of characters, in order to parse the grammatical structure according to the given formal grammar. During parsing, the text is transformed in the data structure, usually in a tree that corresponds to the syntactic structure of the input sequence, and is well suited for further processing. Stemming is the process of reducing the word to the base by rejecting auxiliary parts, such as a suffix. The results of the stemming are sometimes very similar to the definition of the root of the word, but its algorithms are based on other principles. Therefore, the word after processing by the stemming algorithm may differ from the morphological root of the word. Stemming is used in linguistic morphology and in the information search. There are two types of errors in the stemming algorithms: overstemming; understemming. Overstemming is when, during this process, the two words are reduced to one basis, although this should not happen. Understemming is the opposite mistake when words get different bases, where they should have one common. Stemming algorithms try to minimize similar errors, but reducing one type of errors can lead to an increase in the errors of another.

2.2 Analysis of recent research and publications

In [1], the task of developing methods for processing information resources of intellectual Web-systems that were scientifically sound was considered and solved. Based on these methods, technical tools that allowed the creation, distribution and development of e-business systems were developed. To accomplish this task, a study was conducted, the main purpose of which was to determine the peculiarities in the process of processing information resources and finding patterns and dependencies in such processes. After analyzing these factors, we can conclude that there is a contradiction between the active development and the spread of applied information systems and IT in general on the one hand, and a small amount of research on this topic on the other [20, 28-33]. It is precisely because of this contradiction that we can see the existing problem of stalling the innovation development of this sector, which develops much slower than the whole IT sphere, which negatively affects both the growth rates of applied research on this topic. The concept of adaptive ontology, which is used to form mathematical models of the functioning of intelligent systems based on ontologies, was introduced in [4-5]. After analyzing works [4-8, 21-24] in general, one can conclude that the field of development and use of ontologies in the creation of applied systems and research on this topic are actively developing. However, fundamental research on the use of ontologies to make optimal decisions based on the development of information resources in a particular subject area is still absent. Such decisions are based on ontology training methods. Scientific researches that study and develop information processing systems and use the ontologies in the development and operation of information systems in general began at the end of the twentieth century. People and works that have developed the basic theoretical foundations of mathematical models of ontologies include [25-36]:

- J. Salton, T. Gruber, who were the first to suggest using a three-dimensional tuple as a way of representing an ontology;
- N. Guarino, M. Shambard, who in their works identified the ways of development of ontologies, as well as methods of their construction;
- J. Owl, who first introduced such a concept as a conceptual graph;
- M. Montes-Gomez, who used a conceptual graph to represent ontologies.

In the works of R. Knapp, E. Kaufman, I.P. Norenkova, E. Meena, A. Kalli, M. Yu. Uvarova, we can find a description of exactly how to use ontologies in applied solutions. And T. Andreasen, T. Berners-Lee, O. Lazsil, O.V. Palagin, AV Anisimov, considered the problem of building systems that would be based on ontologies, including the problems of working out the Ukrainian language in information resources.

After analyzing the research, carried out by foreign specialists in the field of processing on-line information resources, we can conclude that the main aspects of learning ontologies are [37-45]:

- Assess of the quality of the ontology.
- Building of ontologies, using knowledge derived from heterogeneous sources.
- Integrations of different ontologies within a common subject area.

The main areas of research that are associated with the use of intellectual resources for the development of information resources and ontology training are [46-51]:

- Learning of ontologies using the analysis of natural language texts. [2, 11-13].
- Building of decision support systems that would be based on ontologies [14].
- Development of software tools that allow the development of ontologies in manual and automated mode (OntoEdit, Ontosaurus, Protégé) [15-17].
- Implementation of applications based on ontology and queries to the knowledge base [2, 18].
- Development of ontology description languages (XML, OWL, RDF, DAML +OIL) [19].

2.3 Analysis of the available software products

After looking for available software products, one can come to conclusion that the market for text rubrication is pretty empty. Most solutions are proprietary, or they are designed purely for scientific research. In this case, the quantity of products that would be available to a large mass of people is very small. All found products have a number of common disadvantages: only English is available, bad user interface and lack of storage of the result. One of such products is «uClassify». The lack of languages, other than English, can negatively affect the popularity of the product, since a very large part of the potential audience is not attracted by such a solution.In «uClassify» there is one more significant minus - a low number of rubrics. The small number of rubrics is also a big disadvantage, since the presentation of the rubrics is very general and the rubricating within such a limited number of topics is very superficial. The superficiality of rubrication is bad, because the authors of the content usually write all their articles in one of the global rubric (sports, art, science, etc.) and they are interested in a deeper rubrication of the text, usually performed manually and being very time consuming. The problem with this user interface (Fig. 1) is that it is completely unsuitable for writing large-size articles. That is, the user will have to write his article in a text editor and then copy it in the appropriate field of input on the site. This negatively affects user experience, which significantly reduces the chance that the user will continue to use this software product. Another product is «twinword» company service. This service also supports only English and has a non-convenient user interface (Fig. 2). As in the previous product, the text input field is completely unsuitable for writing the large amount of textual content. It is worth highlighting that even though this site is rubricating better than the previous one, the way to present the result of the rubrication is counter-intuitive and many users may not find the results they need. It is worth noting another common minus both for "twinword" and "uClassify" is the lack of the ability to save results. For people who generate a large amount of text content, it is very important to save this content, as well as the rubrics to which this or that article relates. Consequently, the available systems have flaws regarding the number of languages, the usability, as well as in the ability to store the results.

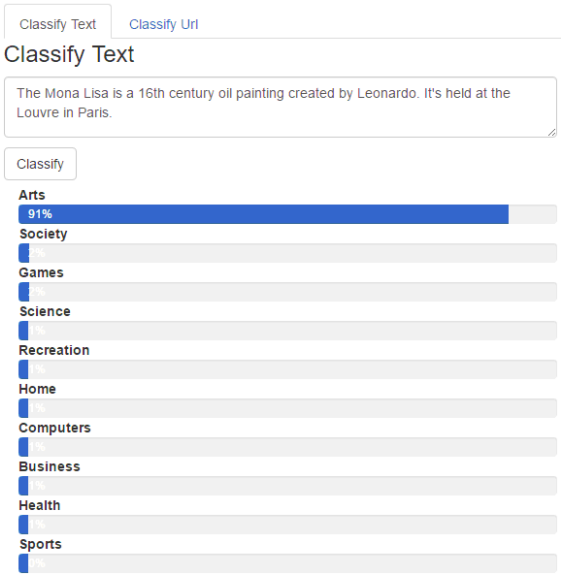

Fig. 1. The user interface of the service «uClassify»

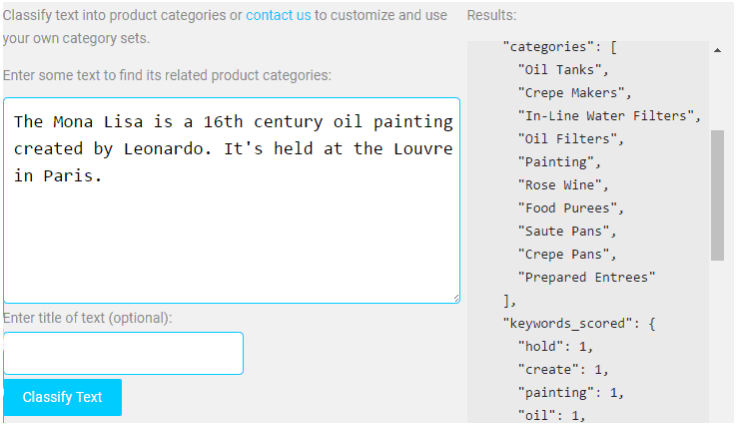

Fig. 2. The user interface of the service «twinword»

3 The object system analysis in the research

The main purpose of the system being developed is the textual content rubrication. We break this goal into smaller goals, which in turn are divided into sub-goals:

1. Interaction with users: authorization, request for rubrics, viewing of the sectioned articles and filtering of articles by categories.

2. Content rubrication: processing of request for rubric, search for entities in the article and analysis of found entities.
3. Defining of rubrics for articles.

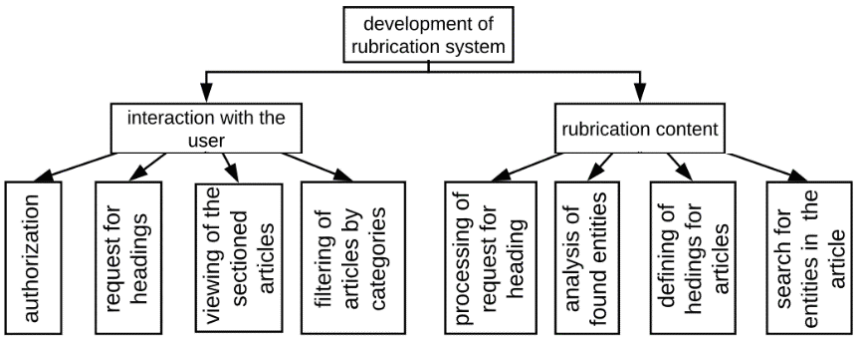

Fig. 3. Tree of objectives for information systems categorizing the textual content

In the developed system, two main external entities can be distinguished: User and Dandelion API. The user has access to his personal profile on the system, and can add his own articles to the system, which will be grouped by the system. Dandelion API is a data source for the system. With this external resource, you can get well structured data from DBpedia, namely the essentials contained in the article being analyzed.

DBpedia is a large crowdsourcing project whose main goal is to structure all the data that is located in Wikipedia. DBpedia itself can be considered the largest ontology in the world with open access. General context diagram of the information system is shown on the Fig. 4. For a better understanding of the work of the system, a decomposition has been made which will help to depict how our system works internally and how data is transmitted between components of the system.

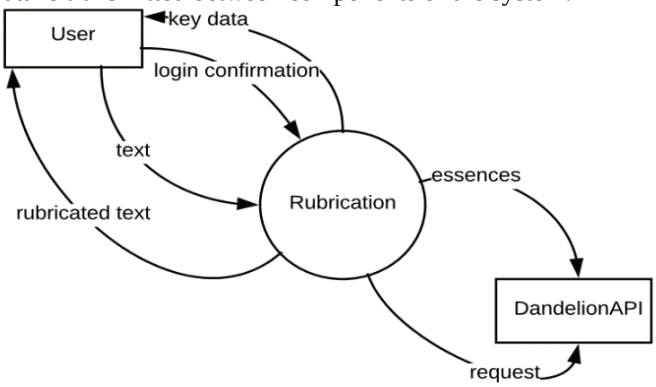

Fig. 4. General contextual diagram

On Fig. 5.rubrics are shown. In the process of elaboration three main functions were selected: user authentication, processing user queries, and analyzing the text. The user authorization feature allows guests to log on to the system. What will allow them to

get into their personal cabinet, and attribute the article to the rubric? The function of processing user queries is responsible for routing all queries between the user and the internal functions of the system. The text analysis function corresponds to both the processing of the text and the connection with the external data source, from where we get the essentials needed for the rubrication of the text.

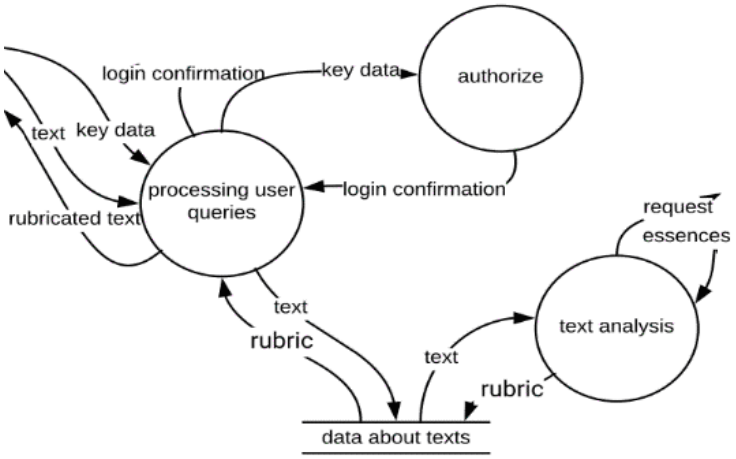

Fig. 5. The detailed diagram

In the text analysis process we can distinguish three main functions: preparing a request, connection with external resources, analysis of entities (Fig. 6).

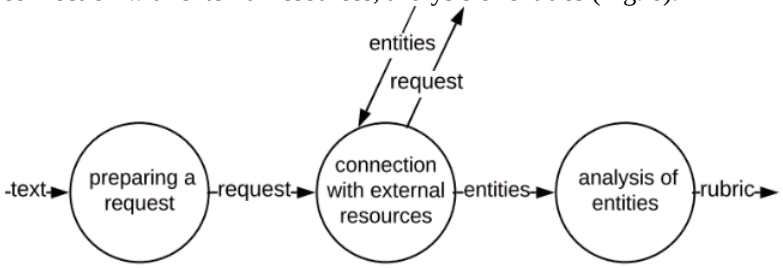

Fig. 6. Detailed text analysis function chart

The request preparation implies sending an inquiry to Dandelion API. The better request is formed, the better results we obtain from the external resource. This function directly affects the quality of our system. This function also directly influences the performance of the system, the better it will work, the more requests the system can handle per unit of time. Next is the entity, which is responsible for the analysis of data received from the external resource and defining categories. This function is the cornerstone of the system. In the process of specification of the query preparation, we get 3 main functions: language definition, text splitting, and installation of additional parameters (Fig. 7). The language definition function is responsible for correctly determining the language of the article, which is very important for the correct analysis of the article.

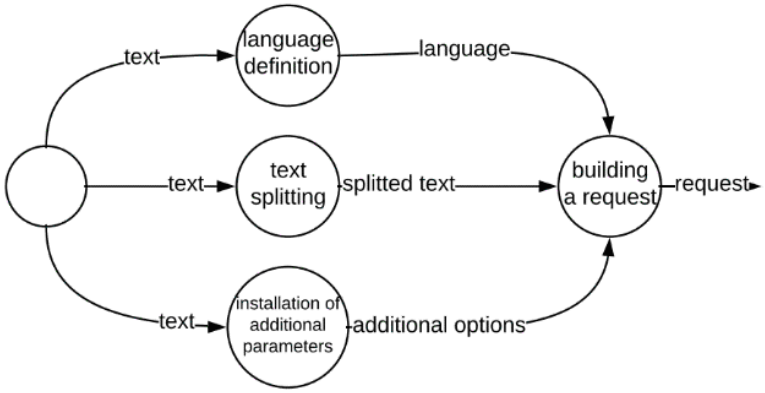

Fig. 7. The chart for request preparation

The text splitting function is responsible for the preliminary preparation of the text before being sent to the external resource. Since the DandelionAPI capabilities are limited, we must adapt the text for the requirements of this resource.

The function of setting additional parameters is responsible for setting additional parameters that the DandelionAPI takes, for example, or when looking for entities to focus more on the context (which is good for a large article) or on the keywords themselves. In the process of specification of the analysis of entities functions, we get three main functions: the definition of the most relevant entities, getting the categories, the definition of the most relevant categories (Fig. 8).

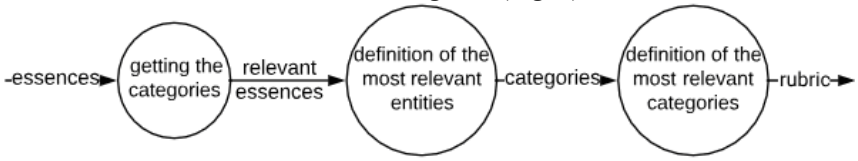

Fig. 8. The chart of entities analysis

The function determining the most relevant entities accepts the DandelionAPI entities and processes them. The function of obtaining categories is responsible for receiving categories of entities, as well as the proper structuring of these categories. The function of determining the most relevant categories takes categories and analyzes them. In the process of analysis is determined, which categories will be the most appropriate for article entered by the user. In the process of specification, we obtain two basic functions: the rejection of entities with low confidence and the rejection of entities that are an alternative name (Fig. 9).

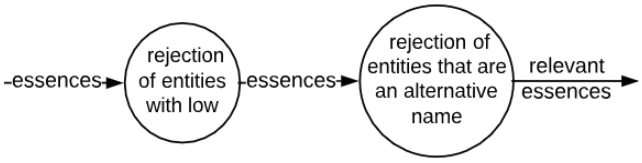

Fig. 9. The chart of the most relevant entities determination

The function of dropping entities with low confidence accepts the DandelionAPI entities and processes them. During the process, the acceptable level of confidence in the relevance to the essence is determined, and all the entities that have the result below this level are rejected. The function of discarding entities is responsible for finding duplicates of the same entities (e.g. auto and car). In the process of elaboration of the function of determining the most rewarding categories, we obtain two main functions: the same categories weighting and the categories rejection with low weight (Fig. 10).

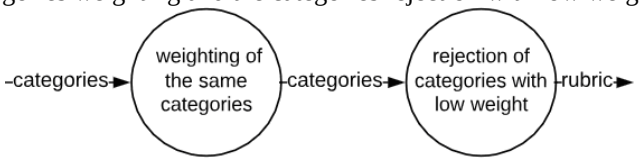

Fig. 10. The chart of entities weighting and selection

The function of weighting the same categories is responsible for the combination of weight categories that were repeated in several entities at the same time.

The category weight function defines the most optimal rated category, and all categories that have the result below are rejected. Figure 11 depicts the hierarchy of tasks of the informational system of textual content rubrication.

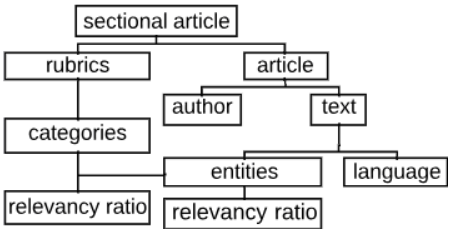

Fig. 11. The hierarchy of objectives for information system categorizing textual content

The root is the main purpose - the rubricated article. To accomplish this goal, we need information about the author of the article, the text of the article and the topics to which it relates. For a proper analysis of the article text, we need to know the language in which it is written and what entities are present in this text. For the proper use of entities we need to know its relevancy ratio for this article. On the other side of the tree, we can see the rubrics that are made of categories. Categories in their turn include entities, and also have their weight.

4 The choice and justification for software platform

The system which categorizes the textual content is best conceived as a web-application. Therefore, the development of this system requires reliable and fast platform that would support Web. Since most modern systems are web-oriented choice is very large.

In order to build an optimal system we need to consider several platforms and select one which fits into our goals. For our system are the following requirements:

- High throughput for client requests.
- Low time serialization data.
- High performance.
- The ability to use different types of servers.
- Easy maintenance.

Among the above platforms were selected Node.js, because the requirements of a particular system it is best suited. To implement the entire system was chosen MERN (MongoDB, Express.js, React.js, Node.js) stack. In fig. 12 can see the overall structure of the system.

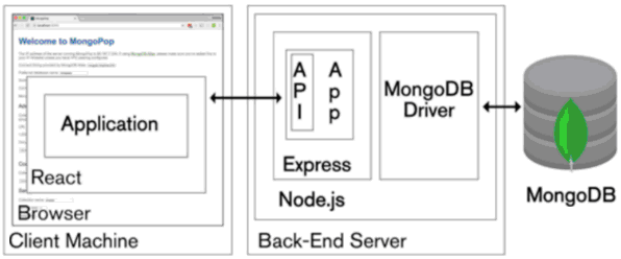

Fig. 12. The scheme of software based on a system MERN-stack

So, on the client side is used React.js, in conjunction with ordinary HTML / CSS. in order to use the latter we utilize the method Babel. Babel - is a compiler that translates any dialect of JavaScript, including CoffeeScript, TypeScript and other add-on language JavaScript ES5, which is supported by almost all browsers including IE8. The power of Babel is in its modularity and extensibility through plug-ins.

To speed up the download of our resources at the request of the client, we will use a minification. This is an approach where the code removed all unnecessary gaps and spaces are removed comments, and big names of variables and methods vary for less.

On the server side is used due Node.js + Express. Node allows the postponed operation records into the database, while continuing to work in this mode, as if these recordings successful. To process client requests for server we use Express.js.

Express is a minimalist and flexible framework for Web applications built on Node.js, providing a wide range of functionality. Having to his disposal many HTTP-assisted methods and intermediate handlers, the robust API can be easily and quickly created. For the database was selected MongoDB. This is document-oriented database (RDBMS). MongoDB occupies a niche between fast and scalable systems that operate on data in the format of a key / value, and relational DBMS, functional and convenient in forming queries. MongoDB has developed a new approach to building database with no tables, charts, queries, SQL, foreign keys and many other things that are inherent in object-relational databases. So the basic architectural idea of the system is the use of isomorphic JavaScript. This means that the whole application is written only in that language. This approach was chosen because JavaScript is ideally suited for this system, since it is asynchronous and non-blocking I / O to allow this system to be used by a large number of users while maintaining high system performance. Also, this approach will allow us to quickly develop a core part of the func-

tionality that will be fully operational, after which developers will eventually be able to easily add new features to the system, in the form of new modules.

5 The description of text categorization software

Figure 13 depicts an "essence-to-link" diagram for the information system of text-based content rubricating. The user writes articles that are read from Essences and are described by Category. In this, the Essences are categorized.

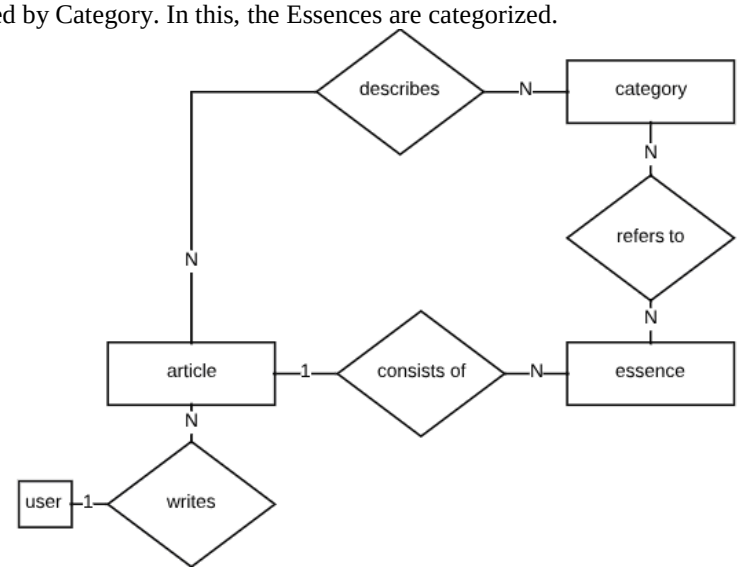

Fig. 13. Chart "entity-relationship" information system for categorization of text content.

To operate the system requires a constant connection to the server. The server is required to connect to the Internet because the system uses external data. The whole system was written using language JavaScript, and is based on technology stack MERN. The user interface is presented in a web page where users may access the system. The purpose of the program is automating the rubricating of text content. This problem is quite important in today's realities, since handwriting is a long process, and correctly-selected rubrics improve both search optimization and the overall experience of using the system. The system being developed also uses the open Dandelion API. The Dandelion API processes receive text, and find the basic essence, and then receives data from DBpedia about the received entity and returns this data to our system. This API is actively developed in DBpedia, and stably supports 7 languages, and more than 50 are in the beta stage. This will allow our system to be used by a large number of people, since it has the ability to articulate text in many languages.

It's worth noting that DBpedia is a structured version of knowledge presented in Wikipedia, which means that we can process articles on a variety of topics. Most systems that deal with text content rubrication have a common problem in that they can categorize articles in just one area of knowledge, but DBpedia has no such prob-

lem. The most important part of the whole process that occurs on the server is processing the received entities. For large text Dandelion API returns a large number of entities and should identify the category that would fit simultaneously to a large number of entities. The task involves the following steps:

1. Determining the language of the article.
2. Breaking the text as DandelionAPI is limited to the number of characters that are sent for processing.
3. Extra options are provided such as entities or when searching for more focus on the context (which is good for large articles) or the keywords themselves. Also, additional parameters may be represented by the maximum number of entities, confidence limit for each entity and country of the author.
4. Sending request to Dandelion API, including entities from the article.
5. Discarding entities with low confidence. During the process of determining the most optimal level of confidence relevant to the essence, and all the entities that have the result below are rejected.
6. Finding and removing entities, which are the alternative name of another entity.
7. Getting the categories of entities.
8. Combining weights for categories, which are repeated in a number of entities simultaneously.
9. Determining the threshold for weight, and all categories that are below the desired result are discarded.
10. Use categories that have reached this step, as a category for the article.

There are many popular sites for blogs / articles, which don't provide an automatic categorization, and the category must be manually put by the authors, or site administration. Using this system, the category will be assigned automatically. In addition to simplifying the life of the author of articles, it will simplify the lives of people who read this article, because in this case, the reader can immediately understand the theme of the article. It can also help in filtering articles by topics. The web – application categorizing the textual content can be used not only in browsers PCs or laptops, but also on mobile browsers. After authentication, the user opens the home page. Above, in the so-called "headers" site posted menu. The menu consists of three buttons: Home; Create - a page for writing; Profile – the private office of user. The system automatically determines the categories of article and assigns them to this article. The user can see both the home page and his private office. After authorization, the user has the opportunity to do the following: view revised articles; filter articles by categories; writing an article for its categorization. See article user to the home page where he can see the cards with basic information about the article, namely, author, subject, category, date of writing (**Error! Reference source not found.** 14). To the left is a filter card where the user can enter the rubrics that interest him, and the system will filter records for these categories (**Error! Reference source not found.** 15).

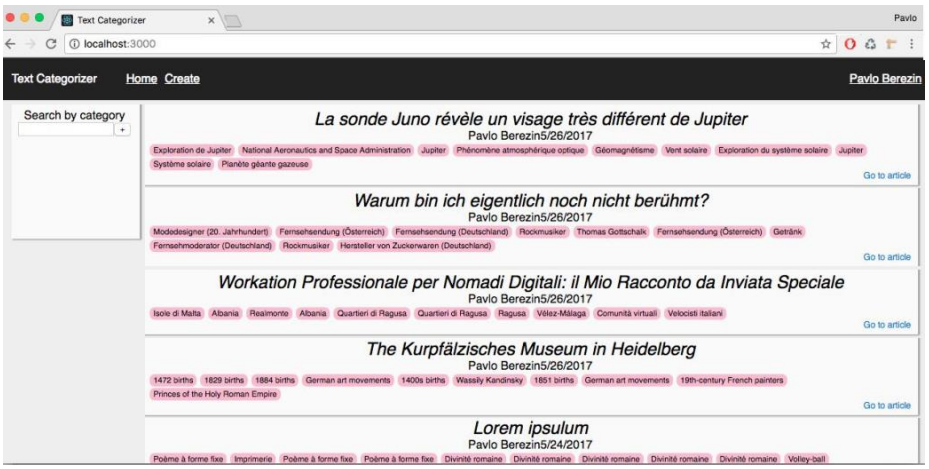

Fig. 14. The home page of application

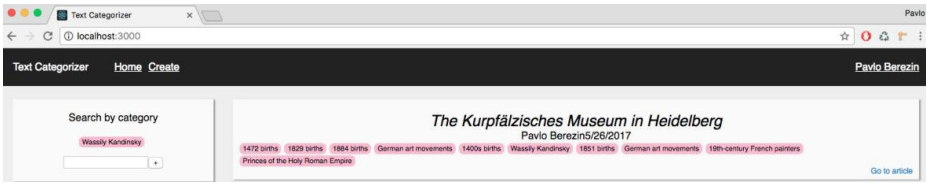

Fig. 15. Simple filter application

To write an article, the user can go to the "Create" tab. This tab has a form with two input fields. In the first field, the author can enter the name of his work. In the second field, the author introduces the text of his work. After that, the author clicks on the "Submit" button and this article is shown by the system and falls into the list of all articles. We consider four very popular language among the authors of the text content: English, German, Italian and French. Each author will work the same way. First, each author must authorize. After that, the author can go to the "Create" page, where he can enter the name and the body of his article. After the system has broken down his article, the author can review its text, sections, and can delete it. First, consider the example of the system in English, German, Italian and French (Fig. 16-19).

Fig. 16. Viewing results of English text categorization

Fig. 17. Viewing results of German text categorization

Fig. 18. Viewing results of Italian text categorization

Fig. 19. Viewing results of French text categorization

6 Conclusion

The implementation of the task of automating textual content rubrication allows processing the large amounts of textual data and filter it before manual analysis by experts. In process of research the key concepts of the topic were studied, and the analysis of recent research and sources was carried out. Also, an analysis and comparison of the available software was conducted, and common deficiencies were found. The system analysis and design of the information system provided the basis for system development. In the course of this analysis, a tree of goals was developed, functional diagrams were also created reflect the data flows inside the system. At the end of the system analysis, a task hierarchy was constructed, which shows what tasks should be executed in order to achieve the final result. As a platform for textual content categorization software, the MERN-stack technology stack was selected. The description of the finished software product described with examples of categorization of texts coming from different languages is provided.

References

1. Alipanah, N., Parveen, P., Khan, L., Thuraisingham, B.: Ontology-driven query expansion using map/reduce framework to facilitate federated queries. In: Proc. of the International Conference on Web Services (ICWS), 712-713. (2011).
2. Euzenat, J., Shvaiko P.: Ontology Matching. In: Springer, Heidelberg, Germany, (2007).
3. Maedche, A., Staab, S.: Measuring Similarity between Ontologies. In: Knowledge Engineering and Knowledge Management, 251-263. (2002).
4. Xue, X., Wang, Y., Hao, W.: Optimizing Ontology Alignments by using NSGA-II. In: The International Arab Journal of Information Technology, 12(2), 176-182. (2015).
5. Martinez-Gil, J., Alba, E., Aldana-Montes, J.F.: Optimizing ontology alignments by using genetic algorithms. In: The workshop on nature based reasoning for the semantic Web, Karlsruhe, Germany. (2008).
6. Calvaneze, D.: Optimizing ontology-based data access. KRDB Research Centre for Knowledge and Data. In: Free University of Bozen-Bolzano, Italy. (2013).
7. Gottlob, G., Orsi, G., Pieris, A.: Ontological queries: Rewriting and optimization. In: Data Engineering, 2-13. (2011).
8. Li, Y., Heflin, J.: Query optimization for ontology-based information integration. In: Information and knowledge management, 1369-1372. (2010).
9. Keet, C.M., Ławrynowicz, A., d'Amato, C., Hilario, M.: Modeling issues & choices in the data mining optimization ontology. (2013).
10. Keet, C.M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R., Hilario, M.: The data mining Optimization ontology. In: Web Semantics: Science, Services and Agents on the World Wide Web, 32, 43-53. (2015).
11. Montes-y-Gómez, M., Gelbukh, A., López-López, A.: Comparison of Conceptual Graphs. In: Artificial Intelligence, 1793. (2000).
12. Biggs, N., Lloyd, E., Wilson, R.: Graph Theory. In: Oxford UP, 1736-1936. (1986).
13. Bondy, J., Murty, U.: Graph Theory. In: Springer. (2008).
14. 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)
15. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)
16. 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)
17. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies, Advances in Intelligent Systems and Computing, 871, 206-221 (2019)
18. 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).
19. 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)
20. 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).

21. 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, 689, 310-319 (2018)
22. Lytvyn, V., Vysotska, V.: Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies, Proc. of the X-th Int. Conf. CSIT'2015, 115-119 (2015)
23. Furgala,Y., Rusyn, B.: Peculiarities of melin transform application to symbol recognition. In: Advanced Trends in Radioelectronics,Telecommunications and Computer Engineering, 251-254. (2018)
24. Kapustiy, B., Rusyn, B., Tayanov, V.: Peculiarities of application of statistical detection criteria for problems of pattern recognition. In: Journal of Automation and Information Science, 37(2), 30-36. (2005)
25. 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)
26. 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)
27. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 343-346 (2018)
28. Rusyn, B., Kosarevych, R., Lutsyk, O., Korniy, V.: Segmentation of atmospheric clouds images obtained by remote sensing. In: Advanced Trends in Radioelectronics,Telecommunications and Computer Engineering, 213-216. (2018)
29. Rusyn, B., Prudyus, I., Ostap, V.: Fingerprint image enhancement algorithm. In: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 193-194. (2001)
30. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, 113-116 (2018)
31. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 330-334 (2018)
32. Kapustiy, B., Rusyn, B., Tayanov, V.: A new approach to determination of correct recognition probability of set objects. In: Upravlyaushchie Sistemy i Mashiny, 2, 8-12. (2005)
33. Varetskyy, Y., Rusyn, B., Molga, A., Ignatovych, A.: A new method of fingerprint key protection of grid credential. In: Advances in Intelligent and Soft Computing, 84, 99-103. (2010)
34. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O.: Uniform Method of Operative Content Management in Web Systems. In: CEUR Workshop Proceedings (Computational linguistics and intelligent systems, 2136, 62-77. (2018).
35. Kravets, P.: The control agent with fuzzy logic, Perspective Technologies and Methods in MEMS Design, MEMSTECH'2010, 40-41 (2010)
36. Davydov, M., Lozynska, O.: Information System for Translation into Ukrainian Sign Language on Mobile Devices. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 48-51 (2017).

37. Davydov, M., Lozynska, O.: Linguistic Models of Assistive Computer Technologies for Cognition and Communication. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 171-175 (2017)
38. 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).
39. Rusyn, B., Torska, R., Kobasyar, M.: Application of the cellular automata for obtaining pitting images during simulation process of thei growth. In: Advances in Intelligent Systems and Computing, 242, .299-306. (2014)
40. Martin, D., del Toro, R., Haber, R., Dorronsoro, J.: Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process. In: International Journal of Innovative Computing, Information and Control, Vol. 5/10(B), 3405-3414. (2009).
41. Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., Rădac, M.-B.: Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy controlled servo systems. In: IET Control Theory & Applications, Vol. 7(1), 99-107. (2013).
42. Ramirez-Ortegon, M.A., Margner, V., Cuevas, E., Rojas, R.: An optimization for binarization methods by removing binary artifacts. In: Pattern Recognition Letters, 34(11), 1299-1306. (2013).
43. Solos, I.P., Tassopoulos, I.X., Beligiannis, G. N.: Optimizing Shift Scheduling for Tank Trucks Using an Effective Stochastic Variable Neighbourhood Approach. In: International Journal of Artificial Intelligence, 14(1), 1-26. (2016).
44. Martin, D., del Toro, R., Haber, R., Dorronsoro, J.: Optimal tuning of a networked linear controller using a multi- In: objective genetic algorithm and its application to one complex electromechanical process. In: International Journal of Innovative Computing, Information and Control, Vol. 5/10(B), 3405-3414. (2009).
45. Nazarkevych, M., Klyujnyk, I., Nazarkevych, H.: Investigation the Ateb-Gabor Filter in Biometric Security Systems. In: Data Stream Mining & Processing, 580-583. (2018).
46. Nazarkevych, M., Klyujnyk, I., Maslanych, I., Havrysh, B., Nazarkevych, H.: Image filtration using the Ateb-Gabor filter in the biometric security systems. In: International Conference on Perspective Technologies and Methods in MEMS Design, 276-279. (2018).
47. Nazarkevych, M., Buriachok, V., Lotoshynska, N., Dmytryk, S.: Research of Ateb-Gabor Filter in Biometric Protection Systems. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 310-313. (2018).
48. Nazarkevych, M., Oliarnyk, R., Dmytruk, S.: An images filtration using the Ateb-Gabor method. In: Computer Sciences and Information Technologies (CSIT), 208-211. (2017).
49. Nazarkevych, M., Kynash, Y., Oliarnyk, R., Klyujnyk, I., Nazarkevych, H.: Application perfected wave tracing algorithm. In: Ukraine Conference on Electrical and Computer Engineering (UKRCON), 1011-1014. (2017).
50. Nazarkevych, M., Oliarnyk, R., Troyan, O., Nazarkevych, H.: Data protection based on encryption using Ateb-functions. In: International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), 30-32. (2016).
51. Babichev, S., Gozhyj, A., Kornelyuk A., Litvinenko, V.: Objective clustering inductive technology of gene expression profiles based on SOTA clustering algorithm. In: Biopolymers and Cell, 33(5), 379–392. (2017)