Learning an Ontology Of Text Data

Vladimir Gorodetsky [1], Olga Tushkanova [2]

[1] *JSC "Evrika", Moskovskiy pr-t., 118, St. Peterburg, 196084, Russia*
[2] *St. Petersburg Federal Research Center of the Russian Academy of Sciences, 14th Line V.O., 39, St. Petersburg, 199178, Russia*

Abstract

The paper describes an algorithm for automating learning an ontology of data represented by natural language texts. Firstly the problem of learning ontology from a sample of texts is formulated. Next, the structure of data ontology that includes the basic level concepts and concepts of higher levels of generalization is described. The algorithm for extracting basic-level ontology concepts, which is based on the use of semantic resources of Wikipedia and DBpedia tools, is presented. Details on the generalization and specialization of basic ontology concepts are given. The new scientific results of the work include the probabilistic model of ontology, the model of interconnections between the basic concepts of ontology and their instance, and the stop criteria for the iterative process of ontology learning.

Keywords

Data ontology, ontology learning, natural language processing.

1. Introduction

The paper describes an algorithm for automating learning an ontology of data represented by natural language (NL) texts. In contrast to the domain ontology, such an ontology can also be called a data sample ontology or a semantic data meta-model.

The use of NL data ontology in modern applications has at least three good reasons. Firstly, since the concepts of ontology have their semantics, a person can easily interpret the results of data processing when solving, for example, machine learning problems in semantically understandable terms. Secondly, the meaning of each word is always approximate, and replacing a word with its synonym only slightly changes the meaning of the text. The third argument for using data ontology is that the transition from describing the text semantics by a set of words to describing it in terms of concept set is a process of data granulation. It is well known that such a process leads to increased stability of computational processes, which is very important, especially for big data. For example, if the data is presented in too much detail in a machine learning task, an effect of overfitting occurs, which leads to instability on new data. Figure 1 illustrates the qualitative influence of the data granulation level on the properties of data processing.

Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference «Integrated Models and Soft Computing in Artificial Intelligence» (IMSC-2021), May 17–20, 2021, Kolomna, Russian Federation
EMAIL: vladim.gorodetsky@gmail.com (A. 1); tushkanova.on@gmail.com (A. 2)
ORCID: 0000-0003-4481-5052 (A. 1); 0000-0001-8394-0783 (A. 2)

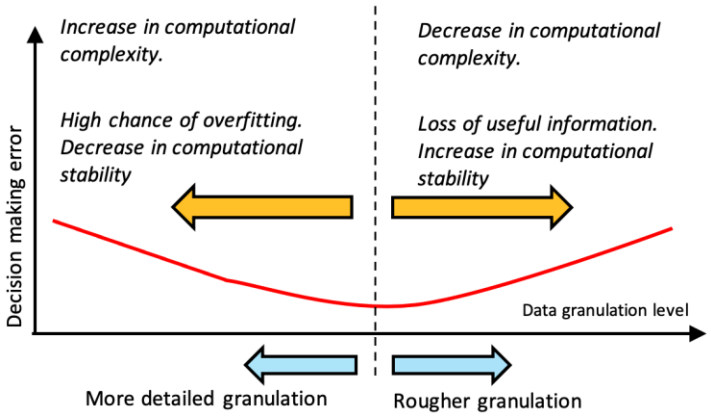

Figure 1: Granulation level and properties of data processing

Summing up the description of the advantages of using data ontology in semantic text processing tasks, we can say that the ontological granulation model of text data is a transition from the description of texts in terms of keywords to its description in terms of semantically interpreted key NL concepts of the ontology. Moreover, this transition makes it possible to implement the automatic learning of the ontology of texts effectively. This paper is devoted to the description of such an algorithm for data ontology building.

2. Problem Statement

Let $\boldsymbol{A} = \{\boldsymbol{A_j}\}$, $\boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{N}$ be a sample of texts. For example, it can be short operational reports of the emergencies service, summarizing everything that happened over the past day in the area of its responsibility. Let this set of texts have been cleared of random errors and outliers and the sample is ready for semantic processing –ontology learning, i.e. for the design of a semantic data structure.

To date, many different technologies have been proposed for formalizing the semantics of text data in terms of ontology concepts. Their detailed analysis is available in [1, 2]. Before describing the proposed algorithm, we will consider the components of the ontology and the structure of their connectivity. Let us recall that the taxonomy of ontology concepts is its mandatory component, which must be, unlike other ontology components. The components of the concept taxonomy together with a dataset and their hierarchy are illustrated in Figure 2.

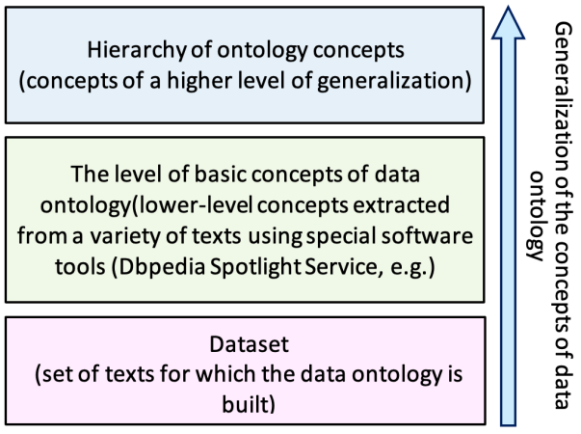

Figure 2: Components of the data ontology and knowledge base

In ontology, two types of concepts should be distinguished – concepts of the basic level and concepts of higher levels of generalization. The concepts of the basic level are constructed directly as a generalization of data ("set up over the data"), and if to account the data level as zeroes level, then the basic concepts constitute the first level of data generalization. The second and subsequent levels of

generalization of ontology concepts are usually constructed sequentially as generalizations of the concepts of the previous levels.

The data ontology together with the data itself and a given structure of inter-level relationships between data instances and basic concepts as well as between concepts of different levels of generalization is usually called a knowledge base. The indicated three-layered structure of the knowledge base (Fig. 2) corresponds to the accordingly layered algorithm for automated ontology learning. At the first stage, a set of basic concepts is found based on a given data sample, and at the next stage, concepts of a higher level of generalization are calculated.

To the present days, various software tools have been used to implement this algorithm. They include, for example, commercial and freely distributed IBM tools (see, e.g., [3]) those involve various lexical databases (for example, WordNet), as well as tools for extracting structured content using Linked Data Web resources [4], among which the most powerful and popular tool is currently DBpedia [5]. The software product, which was developed for automatic extraction of basic-level concepts from a variety of texts, used the DBpedia Spotlight Service tool [6], which uses the DBpedia ontology as a database of concept hierarchy. The ontology concepts extracted in this way are Wikipedia articles structured in a hierarchy of database categories. The following is a description of the ontology generation technology that uses the DBpedia Spotlight Service toolkit.

3. The structure of interlevel connections of ontology

Let us consider the inter-level relationships in the hierarchy of ontology components. Figure 3 shows an abstract example illustrating the proposed organization of relations between the concepts of the base-level ontology and the instances of texts. It assumes that any instance (object, data structure, text, among others) has a reference to the corresponding base-level concept of the ontology. Since each instance of the text may contain words representing different concepts (even a relatively short text may contain dozens or hundreds of representatives of different concepts), each such object (for example, a particular text) refers to a set of concepts, examples of which it contains. In other words, it is assumed that each data instance can be an example containing many basic concepts (Fig. 3) formally defined by a binary relation of type [1:n] between the dataset examples and the basic concepts of the ontology. The explicit setting of this relationship allows to quickly respond to queries to the knowledge base, given in the form of descriptive logic formulas, the arguments of which are examples of concepts.

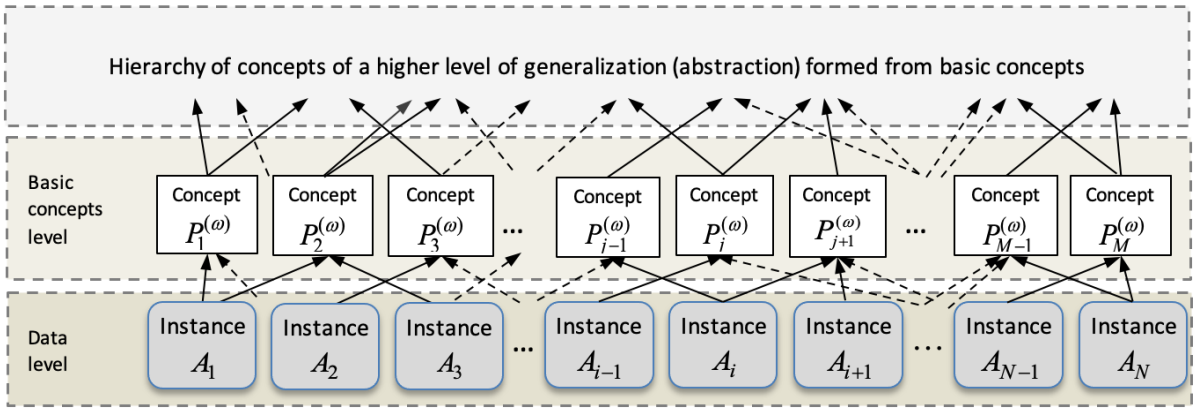

Figure 3: A simplified representation of the data ontology structure

Another assumption about inter-level relationships is that each concept of the basic level of the ontology corresponds to a set of its examples in the dataset, i.e. this relation of the type [1: m] referring from any basic ontology concept to the set of its examples in the dataset. This relationship allows to quickly computing responses to the database queries in terms of descriptive logic formulae, the literals of which are the names of ontology concepts and/or binary predicates of the Tbox scheme. These queries can be similar to the structure of SPARQL queries.

It should be noted that the described pair of relations (specifying relations of the types [1: n] and [1:m]) if they are used together allow to specify relations of the type [:m] and formulate queries

containing both data instances and ontology concepts literals. A schematically presented model of the ontology structure with two-way links is shown in Fig. 3 and 4.

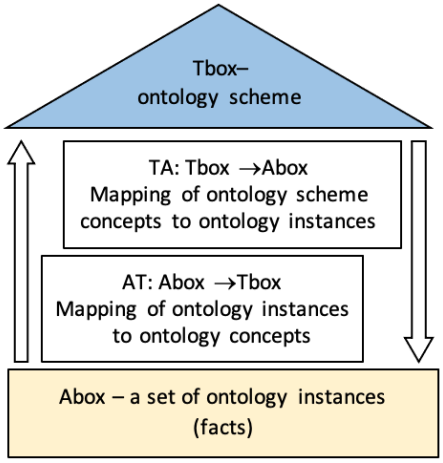

Figure 4: Ontology formal model and its component relations

Let us comment on the practically obtained information about the characteristics of textual datasets and the expected cardinality of the set of basic ontology concepts obtained experimentally. A subset of the ARXIV texts from the Sentence Classification dataset [7] was used to investigate the algorithm for the text data ontology learning. This subset is composed of 100 texts in the .txt format each containing from 400 to 1000 words describing the scientific topic of Machine Learning. We also used a subset of texts from the same set, namely its subset JDM, which contains 100 texts of a similar volume from the journal "The psychology journal Judgment and Decision Making". It turned out that for such a small number of specialized and short texts, the number of basic-level concepts was about 1200.

Let us discuss the question of whether it is realistic to find such two-way connections. Indeed, the practice proved that these connections are constituted during the automated building of the basic level ontology concepts as a part of the total result. If the set of the text data ontology should be expanded further due to getting a batch of new text documents then, to extend ontology according to the novel data, this batch is processed similarly, and, therefore, novel connections between the novel text data instances and ontology concepts of base level is constituted similarly.

Let us pay attention to the advantages of such an organization of inter-level relations. It is well known that the traditional SQL model of data storage in the ontology database is alien to it, and this model is the primary source of hard requirements to the computer resources concerning both memory volume and processing speed. The described model of object data representation in NoSQL format implying direct links from ontology concepts to their instances and vice versa is very natural for executing queries to the ontology, both user queries, and program queries. Indeed, in such a case, there is no need to perform any time and memory-consuming manipulations with a set of SQL tables. A graph database is index-free2, and therefore attaching a NoSQL database to it according to an index-free scheme preserves this property for the whole knowledge base. In the constructed graph knowledge base, the concept examples are the leaves of the knowledge graph general structure.

Another essential advantage of organizing links between examples of concepts and basic-level concepts in the form of a $[n{:}m]$-relation is the following. In this case, instead of the traditional propositional semantics of descriptive logic, predicate semantics can be used, so that the search for answers to queries specified, for example, in the same form as in the SPARQL query language, will be processed not in terms of computationally expensive logical inference, but terms of naturally specified set-theoretic operations [8].

Let us analyze the requirements to the computer resources that the proposed version of the relations between the concept of the basic level ontology and their instances in the database imposes. Let the number of concept examples be counted in the tens of thousands, and the number of concepts of the basic level ontology is of the same order too. Let also text instances be about a thousand words, and the estimate of the average of the number of the links be no more exceed 100 – 200, for each text. For these

[2] Indexing is possible in graph databases, but it is used only to speed up query processing.

assumptions, the cardinality of links can be estimated as several million. There will be the same number of inverse relations. Therefore, the total number of connections to be stored is about a decade of millions, which is not catastrophic. Nevertheless, the memory-related overhead is fully compensated by increasing the speed of solving the query answering tasks. It is important to note that the vocabulary of documents on the same topic should stabilize from day to day so that over time the growth in the number of basic-level concepts should be limited.

4. Algorithm for extracting basic-level ontology concepts

The general scheme of the algorithm for learning ontology of data represented by NL texts is shown in Fig. 5. It assumes that the training sample in the form of a set of texts is prepared in the required form as the input of a block called "Automatic extraction of basic-level ontology concepts from a set of texts". DBpedia Spotlight Service is used as a tool for solving the problem of automatic generation of basic-level ontology concepts. The necessary explanations for its use, as well as documentation, can be found in [6].

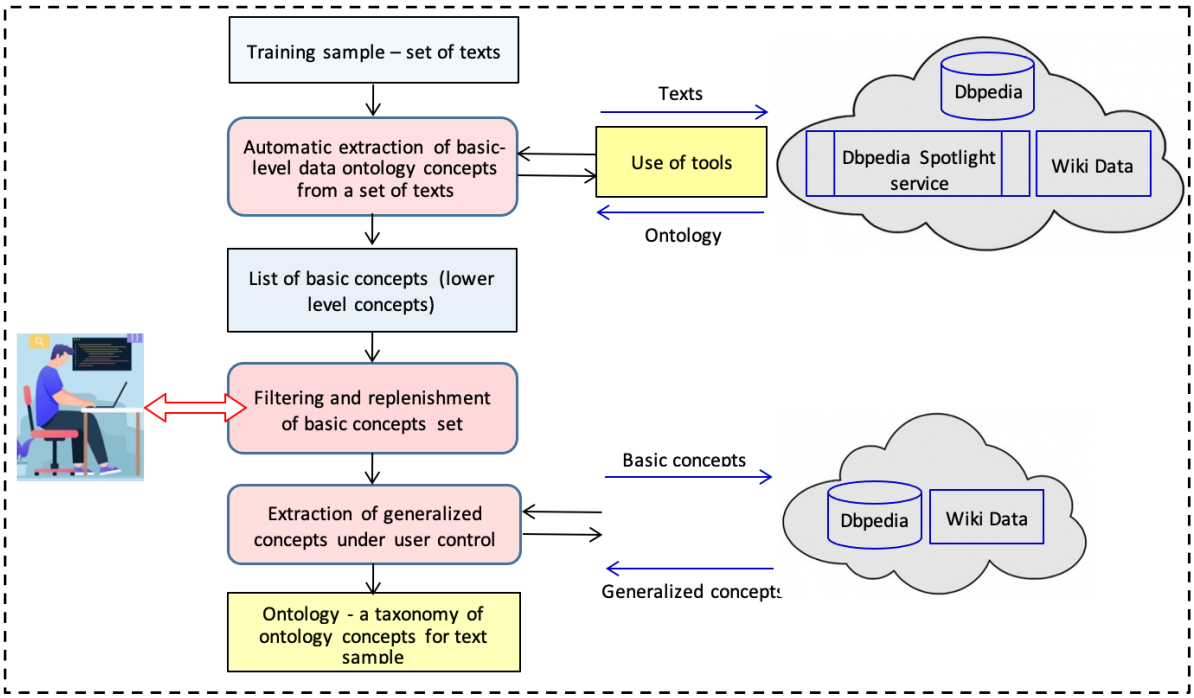

Figure 5: The contour of the automated learning of the text data ontology

So, as a result of text processing using the Spotlight service, the following results will be obtained:

1. Each NL text is assigned a vector of its features - a set of concepts with corresponding significance measure value. It is possible to choose the TF-IDF measure or the measure proposed by the tool authors as a measure of significance. As a result, each text will be represented by a point in the N-dimensional vector space of meanings, where N is the number of found concepts for all the sample texts. The result of such a transformation of text set into a set of points of an N-dimensional space of meanings is usually called Vector Space Model (VSM model).
2. Each found concept is matched to a set of texts in which this concept occurs.

Each found concept is matched to a set of texts in which this concept occurs.

Thus, the solution to the problem will be a set of basic concepts of ontology, their connections with examples and connections of examples with the concepts discussed in section 3, and a VSM model of sample texts.

The operation of the tool is tested on the ARXIV texts sample [7]. The complete set of basic concepts for this data generated using the DBpedia Spotlight Service includes 1061 concepts. Each concept

corresponds to some Wikipedia article and has many examples in the data (texts) in which this concept occurs.

5. The context of data ontology concept

To characterize the properties of each ontology concept, we introduce a data structure called the concept context:

$$Cont\left(P^{(k)}\right) =< P^{(k)}, A^{(k)}, N^{(k)}, p\left(P^{(k)}/A\right) >, \quad (1)$$

where $P^{(k)}$ is the concept of basic level, $A^{(k)}$ is the extent of this concept (a set of concept examples in the dataset), $N^{(k)}$ is the cardinality of the concept extent $A^{(k)}$, $p\left(P^{(k)}/A\right)$ is a sample (a priori) probability of concepts examples $P^{(k)}$ in the text sample A and

$$p\left(P^{(k)}/A\right) = N^{(k)}/N(A), \quad (2)$$

where $N(A)$ is the cardinality of the dataset.

It is important to note that knowledge of the context of each basic concept of the data ontology allows, if necessary, to calculate the extents and other attributes of the contexts of any other concepts that can be obtained by specialization and generalization of the basic concepts without additional scanning of the dataset.

6. Generalization and specialization of concepts

The structure of the data ontology plays a key role in the description of text semantics. Its concepts are formed as a generalization of the basic level concepts. In the developed technology of data ontology learning, an important role also belongs to the concepts resulting from the specializations of the basic concepts that are got at the generalization stage. Therefore, in parallel with the construction of the generalization structure, the structure of the specialization of concepts is also built, which is dual to the structure of ontology concepts.

For generalized concepts of ontology, we will use notations similar to those adopted for basic concepts: we will denote the concepts of the level L with the symbol $\boldsymbol{P}^{(k)}(L)$ and similarly, we will denote the attributes of its context, namely

$$Cont\left(\boldsymbol{P}^{(k)}(L)\right) =< \boldsymbol{P}^{(k)}(L), \boldsymbol{A}^{(k)}(L), N^{(k)}(L), p\left(\boldsymbol{P}^{(k)}(L)/\boldsymbol{A}\right) >. \quad (3)$$

The concept that is dual to the concept $\boldsymbol{P}^{(k)}(L)$ is denoted by $\boldsymbol{P}^{(k)}(L)$. The context corresponding to the dual concepts is introduced similarly:

$$Cont(\ \boldsymbol{Q}^{(k)}(L)) =< \boldsymbol{Q}^{(k)}(L), \boldsymbol{B}^{(k)}(L), M^{(k)}(L), p\left(\boldsymbol{Q}^{(k)}(L)/\boldsymbol{A}\right) >. \quad (4)$$

In this formula, the elements located at a certain position have the same meaning as the elements in formula (3) in the context of the corresponding concept of ontology. Probability value $p\left(\boldsymbol{Q}^{(k)}(L)/\boldsymbol{A}\right)$ in the context of the dual concept of ontology is calculated by a formula similar to the formula (2):

$$p\left(\boldsymbol{Q}^{(k)}/\boldsymbol{A}\right) = M^{(k)}/N(\boldsymbol{A}). \quad (5)$$

Thus, two structures are built over the set of basic concepts – the structure of generalization of concepts and the dual structure of specialization. The generalization structure is built using the DBpedia Spotlight Service, and the dual structure is built automatically by simply calculating the context components $\boldsymbol{B}^{(k)}(L)$, $M^{(k)}(L)$ и $p\left(\boldsymbol{Q}^{(k)}(L)/\boldsymbol{A}\right)$ according to the obvious formulas.

For the algorithm of iterative generalization of the ontology basic concepts, the problem of stop criteria arises because, without it, an ontology may get a catastrophically large number of concepts. For example, for Amazon data [9], the technology supported by DBpedia tools leads to a hierarchy of concepts containing 24 levels, and the resulting ontology may contain too many concepts exceeding the number of words in the set of texts based on which this ontology is built. It is clear that for practical purposes, such an ontology is not only unnecessary but also harmful from the standpoint of computational efficiency. Therefore, we need to solve the problem of stop criteria.

The authors propose two semantic stop criteria formulated in the form of the rules below. The first of them is based on a comparison of the contexts of a pair of ontology concepts connected with the immediate following relation. Let us explain its essence with an example. Let there be two concepts in the constructed fragment of the ontology, for example, Dynamic Systems and System theory, while the

second concept is a generalization of the first. Let both concepts have the same extent. If the System theory generalization for the concept of Dynamic Systems is added to the ontology, then the extent of the generalized concept in this pair of concepts will be the same as that of the predecessor concept, and therefore it does not add additional meaning to the semantics of the text to the ontology. Moreover, if we introduce the generalized concept of System theory into the ontology and continue the process of generalizing the concept of Dynamic Systems, then a large number of new, more general concepts with absolutely the same extent may appear, since any more general concept always contains the extent of a more special concept. This argument allows us to formulate Rule 1 of stopping the generalization process:

Rule 1. For any pair of concepts of the ordinal structure of the data ontology $\boldsymbol{P}^{(k)}(L)$ and $\boldsymbol{P}^{(r)}(L+1)$, such that $\boldsymbol{P}^{(k)}(L) < \boldsymbol{P}^{(r)}(L+1)$ (comparable in the order of generalization), which have the same extent, i.e. $\boldsymbol{A}^{(k)}(L)$=$\boldsymbol{A}^{(r)}(L+1)$, the ontology is included only the smaller of them, i.e. the concept of $\boldsymbol{P}^{(k)}(L)$, in our case.

The second criterion is derived from the relationship between the properties of ontology concepts and their dual structural elements constructed over the same set of basic concepts. Suppose a concept appears in the data ontology that generalizes, for example, a pair of concepts that do not have common elements in their extents. In that case, it means that independent sets of examples correspond to them, and the concept that generalizes them does not contain new information about the relationships of these concepts. In this case, the dual concept for such a generalization will have an empty extent, which can be used as a stop criterion:

Rule 2. If dual-element $\boldsymbol{Q}^{(k)}(L)$ of the concept specialization structure for a concept $\boldsymbol{P}^{(k)}(L)$ has empty extent, i.e. $\boldsymbol{B}^{(k)}(L) = \phi$, the concept $\boldsymbol{P}^{(k)}(L)$ and all greater concepts are not included in the data ontology.

Let us now shortly describe an iterative algorithm for constructing a data ontology at the generalization stage. In it, at each iteration, a set of ontology concepts represented by their contexts is used as source data:

$$\{Cont\left(\boldsymbol{P}^{(k)}\right) =< \boldsymbol{P}^{(k)}, \boldsymbol{A}^{(k)}, N^{(k)}, p\left(\boldsymbol{P}^{(k)}/\boldsymbol{A}\right) >\}, k = 1, 2, \dots, n(k). \quad (6)$$

The basic concepts form the first level of the target data ontology. Note that the concepts of ontology and the dual concepts of the specialization structure are the same at this level.

Starting from the already built set of basic concepts, the Spotlight Service matches each found concept with the URI of the Wikipedia article. Therefore, it is possible to get its parent categories and then search for possible generalizations from them. The following is a more formal description of the iterations:

Step 1. For the set of ontology concepts to be generalized, the generalization step is performed using the DBpedia Spotlight Service. For each new concept, the values of the attributes of its context and the context of the dual concept are calculated.

Step 2. If there exist concepts with empty extents among the newly found dual concepts then the generalized ontology concepts corresponding to them are marked with a service symbol that informs the program that these concepts are not subject to further generalization.

Step 3. If among the newly constructed concepts there are those for which the condition of Rule 1 is met then these concepts are marked with a service symbol that informs the program that this concept is not subject to further generalization.

Step 5. If the set of ontology concepts that are the subjects to generalization which are not marked with a service symbol is not empty (i.e. there exist some concepts that allow further generalization), then the number of the generalization step increases ($L = L + 1$) and the algorithm goes to the beginning of step 1.

The end of the algorithm.

7. Conclusion

The paper proposes and experimentally analyzes an algorithm for automating learning an ontology of an NL text dataset. The algorithm is based on the use of semantic resources of Wikipedia and DBpedia tools. The developed software package is registered in [10].

The new scientific results of the work include the following:

1. The probabilistic model of ontology that is its important property in the subsequent machine learning procedures.
2. The model of interconnections between the basic concepts of ontology and their instances; this model allows increasing the speed of query processing significantly.
3. The stop criteria (stopping rules) for the iterative process of ontology learning, which, as experiments have shown, can significantly increase the speed of its learning and exclude concepts that do not contain any additional information about relations on a set of concepts of NL texts. The algorithm can be applied to the tasks of incremental ontology learning.

Further research is planned to focus on the learning of an ontology of text data in decision-making tasks in situations where the samples of text data are small or even absent [11].

8. References

[1] V. Gorodetsky, V. Samoylov, O. Tushkanova, Agent-based Customer Profile Learning in 3G Recommending Systems: Ontology-driven multi-source cross-domain case, in: L. Cao et al. (Eds.), Agents and Data Mining Interaction, vol 9145 of Lecture Notes in Computer Science, Springer, Cham, 2014, pp. 12-25. doi: 10.1007/978-3-319-20230-3_2.

[2] V. Samoilov, O. Tushkanova, Knowledge Net: a model and system for the accumulation, presentation and use of knowledge and data, Ontology of Design 1 (31) (2019) 117-131. doi: 10.18287/2223-9537-2019-9-1-117-131.

[3] IBM-Watson, 2021. URL: https://www.ibm.com/watson/natural-language-processing.

[4] Synthesis Lectures on the Semantic Web: Theory and Technology, 2021. URL: https://www.morganclaypool.com/toc/wbe.1/1/1.

[5] DBpedia - Global and Unified Access to Knowledge Graphs, 2021. URL: https://wiki.dbpedia.org.

[6] DBpedia Spotlight Shedding light on the web of documents, 2021. URL: https://www.dbpedia-spotlight.org.

[7] Sentence Classification Data Set, URL: https://archive.ics.uci.edu/ml/datasets/Sentence+Classification.

[8] V. Gorodetsky, O. Tushkanova, Effective Big Data Processing Techniques for Decision Making, in: Digest of plenary reports of the Russian conference "Intelligent control systems", St. Petersburg, 2016.

[9] R. He, J. McAuley, Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering, in: Proceedings of the 25th International Conference on World Wide Web - WWW '16, 2016. doi:10.1145/2872427.2883037.

[10] O. Tushkanova, Component for automatic extraction of data ontology based on semantic analysis of concepts, 2019. Patent No. 12 RU 2019615010345, Filed April 4th., 2019, Issued April. 17th., 2019.

[11] Y.Q. Song, S. Upadhyay, H.R. Peng, S. Mayhew, D. Roth, Toward any-language zero-shot topic classification of textual documents, Artificial Intelligence 274 (2019) 133-150. doi: https://doi.org/10.1016/j.artint.2019.02.002.