Analysis of Scientific Texts by Semantic Inverse-Additive Metrics for Ontology Concepts

Viktor Hryhorovych

Lviv Polytechnic National University, S. Bandera street, 12, Lviv, 79013, Ukraine

Abstract

Semantic analysis of textual information is a problem that does not lose its relevance. It has to be solved when solving such tasks as automation of filtering, classification, and clustering of text documents, automation of abstracting a given text, automation of evaluation of answers to open test tasks, automatic construction of a semantic network for a given text, etc.

All such tasks are limited to quantifying the elements of a text document and the relationships between them.

This paper proposes a method of semantic analysis based on inverse-additive metrics, which takes into account the semantic distance between the terms of the ontology in the text document being analyzed. This metric allows you to correctly process cases where there are several paths in the oriented graph of the ontology from one concept node to another.

Semantic analysis of scientific documents is considered, as such texts have a clear structure. The concept of semantic distance between the terms of a scientific text and the semantic weight of a scientific text is introduced. The semantic weight of individual fragments of a scientific text is used to solve the problem of automatic abstracting.

Keywords 1

semantic analysis, semantic metrics, ontology, semantic distance, semantic weight, automatic abstracting, text analysis

1. Introduction

Semantic analysis of textual information is a problem that does not lose its relevance. It has to be solved when solving such tasks as automation of filtering, classification, and clustering of text documents, automation of abstracting a given text, automation of evaluation of answers to open test tasks, automatic construction of a semantic network for a given text, etc.

All such tasks are limited to quantifying the elements of a text document and the relationships between them.

Many quantitative characteristics approaches are essentially parsing because they use a frequency approach based on the number of occurrences of keywords. Some techniques use the conversion of texts into real number vectors and the use of the mathematical apparatus of vector algebra to quantify the corresponding text documents. Such approaches are also reduced to the number of occurrences of certain keywords or comparison with a template – the basic body of textual information.

This paper will use semantic analysis based on an inverse-additive metric that takes into account the semantic distance between ontology terms in the text document being analyzed. This metric allows you to correctly process cases where there are several paths in the oriented graph of the ontology from one concept node to another.

Semantic analysis will be considered for scientific documents, as such texts have a clear structure. To do this, introduce the concept of semantic distance between the terms of a scientific text and the semantic weight of a scientific text. The semantic weight of individual fragments of a scientific text will be used to solve the problem of automatic abstracting.

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland
EMAIL: viktor.grigorovich@gmail.com; viktor.h.hryhorovych@lpnu.ua (V. Hryhorovych)
ORCID: 0000-0002-5828-067X (V. Hryhorovych)

At the same time, it is necessary to overcome some difficulties in the implementation of the proposed approach, associated with critical nodes on the way in the oriented graph of the ontology from one concept node to another.

2. Related Works

Both works on philology and scientific articles in the field of information technology and computational linguistics are devoted to the semantic analysis of texts.

The first category includes works [1-2]. In [1] literary terms are investigated. This is a philological study that analyzes terms related to the theory of literature, its history, processes and dramatic works. Work [2] is a philological study that combines approaches that consider the text as a set of communicative blocks, zones of compression and scattering, noun chains, and thematic progression.

The following work concerns the development and application of information technology for the semantic analysis of texts. In [3], a method of latent semantic analysis is described, which assumes that words close in meaning will occur in similar fragments of text. A matrix containing the number of words per document is considered (rows represent unique words, and columns represent each document). The documents are compared based on the cosine of the angle between two vectors (or the scalar product between the normalizations of two vectors) formed by the corresponding two columns. Values close to 1 represent very similar documents, while values close to 0 represent very different documents. The work [4] describes the results of three experiments that demonstrate the use of methods of latent semantic analysis for the study of texts: the correspondence of annotations to annotated texts; characterization of essay quality and measurement of text coherence. In [5] the research of scientific texts on the basis of the constructed annotated corpus of 8736 citations is described. 6 different algorithms of machine learning with preliminary normalization of data for noise removal were used. In [6] described intelligent information systems for semantic analysis, semantic interpretation, and understanding of data, designed to support data management processes. These processes are performed using linguistic techniques and semantic interpretation of the analyzed sets of information/data during the processes of description and interpretation. Methods of semantic interpretation allow extracting information from the sets of analyzed data. This improves decision-making processes and improves the entire data and information management process. In [7] one of the approaches and methods of semantic analysis is considered - the approach based on vocabulary. It consists in calculating the orientation of sentiments of the whole document or set of sentences based on the semantic orientation of vocabulary. In [8], a set-theoretic approach is proposed to describe the double relation “M is a model of system S”. In [9] new generation systems are studied – cognitive systems. Their feature is the semantic analysis of data. Cognitive information systems, their definitions, discussion of perception models, classification of cognitive information systems, and presentation of decision-making methods are discussed. Particular attention is paid to the decision-making process in cognitive systems. In [10] the method of automated identification of metaphors in the semantically annotated corpus of texts is described. Work [11] is the first in its field. It describes the author's method of using NLP and semantic analysis of texts for mapping supply chains. In [12] the developed method of semantic analysis for processing Ukrainian-language texts is described, which allows analyzing Ukrainian-language content using the method of latent-semantic analysis (see [13] and [14]) and morphoanalyzer Pymorphy2. The NER model was used to expand the semantic capabilities of the developed system. [15] describes the results of studies of phenomenology and the concept of unambiguity in linguistics when comparing the same aspects of accuracy. The theory of semantic states and the apparatus of hyperchains in lexicographic structures were used, which made it possible to formalize the semantics of language constructions and to distinguish between the concepts of accuracy and unambiguity. The concept of accuracy is introduced as a definition of all lexical meanings, semantic states, and their superpositions in which the analyzed token can function. [16] describes the transformation of words into vectors of real numbers (embedding words, see [17] and [18]). Previously, no vector research was conducted using the Word2vec technique to create a Ukrainian word corpus. Libraries of licensed open-source software libraries "Gensim" were used to implement machine learning using Word2vec methods in Python and calculations of cosine affinity of the obtained vectors. The extent to which vectors are obtained from the Ukrainian corpus and how

word vectors are grouped and associated according to the morphological features of Ukrainian language suffixes have been studied. [19] describes the use of generating grammars in linguistic modeling. To automate the study and synthesis of natural language texts, sentence syntax analysis is used. The main differences in the grammatical and phonetic structure of English and Ukrainian languages are analyzed. The optimal method of automatic processing of the text set of Ukrainian - language content in relation to essential keywords and identification of content categories, analysis of syntax, and semantics of the text is determined. Article [20] defines the specification language for high-level testing scenarios for testing critical systems based on the built-in Uppaal Timed Automata model. The scalability of the method is demonstrated by the example of satellite software testing.

[21] describes the web system developed by the authors to visualize the structure of the ontology data. The creation of the system of visualization of ontologies of the subject area on the example of ontology models is described in detail. The system provides tools for a dynamic display of different types of ontographs in accordance with the established visualization criteria, which allows their use in information systems for the operational management of objects. creation of a system of visualization of ontologies of the subject area on the example of ontology models. Here is an example of visualizing the concept of "computer network attack". [22] describes the developed system of production environment management, which simplifies the process of analyzing a large amount of information from different sources. [23] describes the results of a study of the process of forming a semantic core for a web resource. The study expands the concept of the semantic network based on four components: URI, ontology, data, and semantic language. This concept is implemented in the work with the help of the semantic core. The core is formed on the principle of annotation based on an algorithm based on the semantic network and the method of Data Mining technology. Thus, an alternative implementation of Semantic Web components is proposed. The RDF scheme is used to represent the semantic core. The software is implemented using JavaScript using the Node JS library. [24] describes an approach based on the fact that ontology is a mechanism for obtaining information on the Internet in a more structured way using the semantic network. The focus is on choosing the presentation of documents suitable for creating user-profiles and supporting the content-based search process. The semantic web solves this problem, makes data understandable by machines in the form of an ontology, and the multi-agent extracts useful knowledge hidden in this data and makes it available. [25] describes the ontological decision support system in automated military control systems. The core of the system is an ontology that combines three levels: 1) an ontology focused on the domain, subject area - contains the concept of taxonomy, relationships, instances of classes, and different types of constraints - axioms. Axioms establish semantic rules for the system of relations; 2) task-oriented ontology - describes the solution of specific problems, contains knowledge of the specifications of structures (databases) and methods of data processing; 3) ontology of the upper level - describes the categories - the concept of the upper level. Examples are physical, functional, and behavioral concepts and attitudes that relate to general scientific concepts. A decision support system has been developed - a prototype of an automated control system for the Land Forces of the Armed Forces of Ukraine according to the standards of NATO member countries. In [26] the authors describe their unique technology of organizing data warehouses on the basis of consolidated data from libraries, archives, and museums. The technology is based on multidimensional data analysis and building a data hypercube. This technology is interesting because in combination with the semantic analysis of textual information will simplify and increase the efficiency of the social and communication environment in general and information processing technologies in it. [27] describes the developed system of automated compilation and formation of digests of electronic publications in the media, the selection of critical content from one or more documents, and the formation of concise reports based on them. The system monitors information, receives large amounts of data, analyzes, organizes data using an automatic header, collects information, indexes material and stores it in a database, solves thematic filtering, and generates digests automatically. [28] describes the developed unified methodology for processing information resources in e-content commerce systems. A formalized method of content analysis is used, which allows you to fully automate the process that occurs when an author adds a new article. The method identifies articles whose topics are similar to those viewed by the user.

3. Methods

3.1. Inverse-additive metric for ontology concepts

The inverse-additive metric [29] allows calculating the distance between ontology concepts in the case when there are several paths from one concept to another. Consider the representation of ontology concepts and the relationships between them in the form of an oriented graph. Then each concept will correspond to a certain node. If the ontology is organized in the form of an explanatory dictionary, then each term is a keyword and its interpretation; and the text of the interpretation contains keywords - references to other terms. This is the reason for the existence of several paths from one node to another in the oriented graph of the ontology.

Define the distance $R(A, B)$ between the concepts A and B as follows:

$$\frac{1}{R(A,B)} = \sum_{i=1}^{K} \frac{1}{N_i}, \quad (1)$$

where

N_i – is the number of transitions from concept A to concept B on the i-th path, i=1, …, K,

K – is the number of different paths that can be taken on the oriented graph of a particular ontology from concept A to concept B.

If there is a single path between concepts A and B, then the distance between them is equal to the number of transitions from one concept to another:

$$R(A,B) = N \quad (2)$$

The more paths there are between concepts, the smaller the distance will be, that is, the semantically closer the corresponding terms will be.

It is proved that this definition satisfies the axioms of the metric. Note that a pair of complementary symmetric connections must be introduced for the axiom of symmetry. For example, for an explanatory dictionary ontology, it is a pair of "uses-of" - used-in relationships, which allows the symmetry axiom to be met for the proposed metric in the following interpretation:

$$R_{used-in}(A,B) = R_{uses-of}(B,A) \quad (3)$$

3.2. Semantic distance between terms of a scientific text

Scientific texts have a clear hierarchical structure, as shown in Fig. 1. Here:

Level 1 - the name of the document (root node);

Level 2 - authors, keywords, abstract, sections, list of sources used. Information content of this level: list of authors, list of keywords, the text of the annotation, titles of sections, names of used sources;

Level 3 - sections. Information content of this level: the names of sections. Other levels are possible.

Level N - sentences. Information content of level N: words that are part of one sentence.

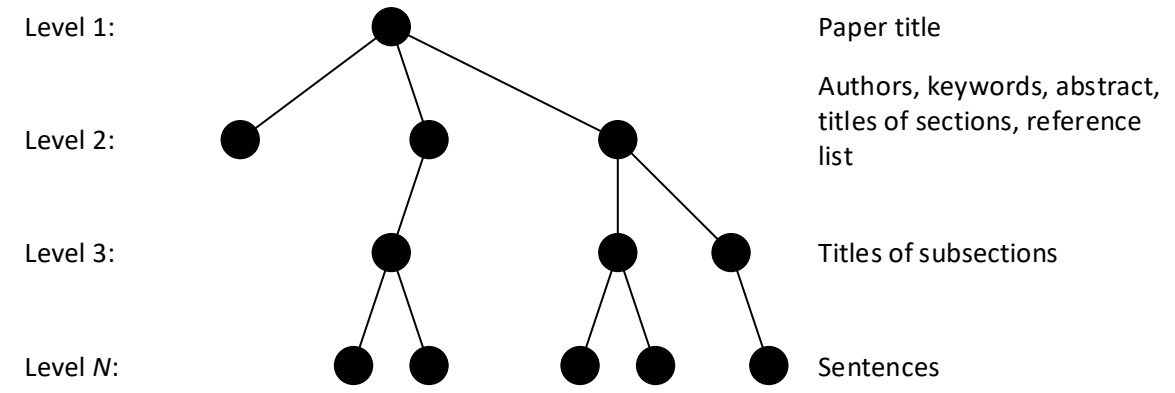

Figure 1: Hierarchical structure of scientific texts

Consider the problem of calculating the semantic distance between two terms *A* and *B* from the ontology, which are in some text. That is, the ontology contains these terms as concepts *A* and *B*. The distance *R*(*A*, *B*) between these concepts in the ontology is determined by the formula (1).

The distance between two terms in a scientific text should be defined to take into account the following. 1) Repeated occurrence of each of these terms in the scientific text. 2) The hierarchical structure of the document. 3) The presence of many paths from each occurrence of the term *A* to each occurrence of the term *B*. 4) The distance between the corresponding concepts of these terms in the ontology.

$$\frac{1}{R(A,B)} = \sum_{i=1}^{N_A} \sum_{k=1}^{N_B} \frac{1}{R(A_i, B_k)}, \tag{4}$$

where

$R(A,B)$ – is the semantic distance between the terms *A* and *B* from the ontology in the scientific text;

N_A – the number of occurrences of the term *A* in the scientific text;

N_B – the number of occurrences of the term *B* in the scientific text;

$R(A_i, B_k)$ – is the semantic distance between A_i – the *i*-th instance of the term *A*, and B_k – the *k*-th instance of the term *B* from the ontology in the scientific text.

It is necessary to consider in what places of the text there are the specified terms.

Consider the example shown in Fig. 2. Suppose that in the ontology used to evaluate a scientific text, the distance between concepts *A* and *B* is *L*.

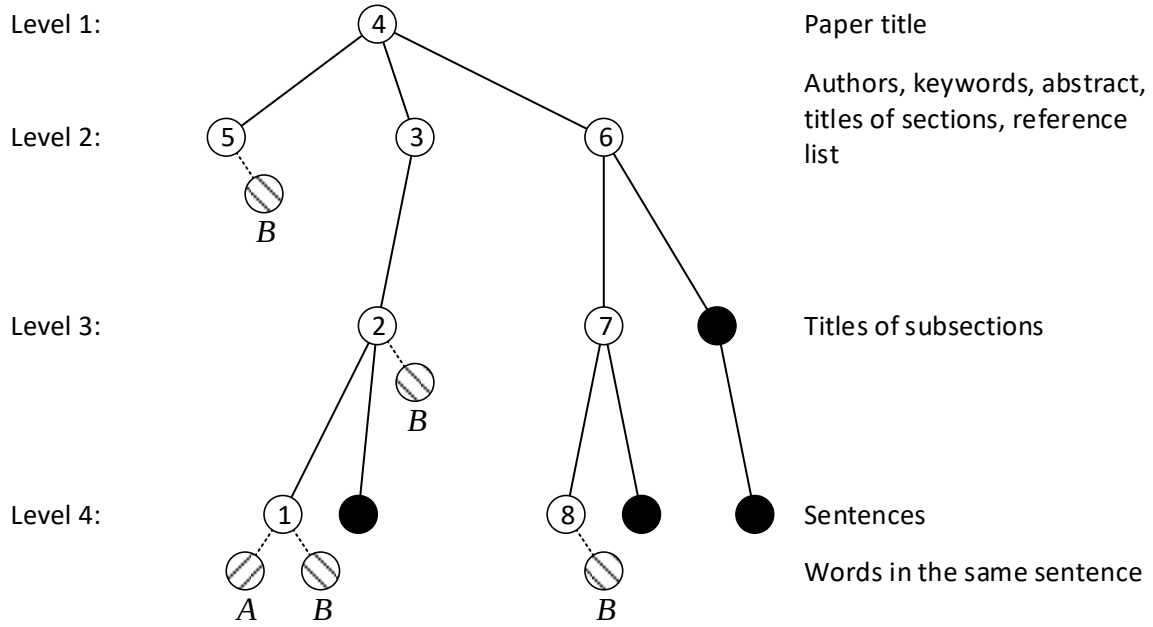

Figure 2: Inclusion of terms *A* and *B* in the scientific text

Assume that in the text under study, the term *A* (the keyword of concept *A*) is contained only in sentence 1 (see Fig. 2). Suppose also that the term *B* (concept keyword *B*) is contained in the same sentence 1, the title of subsection 2, the list of keywords 5, and the sentence of another section 8 (Fig. 2). Thus, there are $K = 4$ different paths from term *A* to term *B* in the graph of the hierarchical structure of this text:

$A\ (1) \rightarrow B$ – distance = L

$A\ (1 \rightarrow 2) \rightarrow B$ – distance = $L + 1$

$A\ (1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5) \rightarrow B$ – distance = $L + 4$

$A\ (1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8) \rightarrow B$ – distance = $L + 6$

According to formula (1):

$$\frac{1}{R(A,B)} = \frac{1}{L} + \frac{1}{L+1} + \frac{1}{L+4} + \frac{1}{L+6}$$

Thus, the more occurrences of ontology terms in a scientific text, the smaller the semantic distance between them.

3.3. Semantic weight and semantic size of a scientific text

For the semantic analysis of scientific texts, it is necessary to define the concept of semantic weight. The semantic weight *SW* of a scientific text relative to a certain ontology is the sum of the inverse distances between all terms of the ontology in the scientific text:

$$SW = \sum_{A \neq B} \frac{1}{R(A,B)}, \quad (5)$$

where

$R(A,B)$ – is the semantic distance between the terms *A* and *B* from ontology in a scientific text (4).

Thus, the more terms there are in a scientific text (or its fragment) and the smaller the semantic distance between them - the greater the semantic weight of this text (fragment).

The inverse value can be considered as the semantic size of a scientific text (then the semantic size will have the same dimension as the semantic distance).

3.4. Automatic abstracting of a scientific text based on the semantic weight of sentences

Semantic analysis of scientific texts based on semantic weight (5) will solve the problem of automatic abstracting based on the selection of sentences (paragraphs) with the highest semantic weight.

To do this, first define the sentence $s_{max\,SW}$ with the largest semantic weight:

$$s_{max\,SW} = \arg\max\{SW(s)\},$$
$$SW(s) = \sum_{A_s \neq B_s} \frac{1}{R(A_s,B_s)}, \quad (6)$$
$$A_s \in s, B_s \in s,$$

where

$R(A_s,B_s)$ – is the semantic distance between the terms A_s and B_s from the ontology in the sentence *s* of the scientific text (4).

$SW(s)$ – is the semantic weight of the sentence *s*.

Next, we determine the degree of compression μ in automatic abstracting: the result of S_{result} will include those sentences whose semantic weight differs from the maximum by no more than μ:

$$S_{result} = \{s | SW(s) \geq (1-\mu) SW(s_{max\,SW})\}, \quad (7)$$

where

S_{result} – set of sentences, the result of automatic abstracting;

$SW(s_{max\,SW})$ – the maximum semantic weight of a sentence in a scientific text.

Thus, automatic abstracting will be implemented as a selection of sentences and compression of the scientific text in $1/\mu$ times relative to the initial number of sentences based on their semantic weight.

4. Experiment

4.1. Calculation of the distance between the concepts of ontology

The algorithm for calculating the distance between two concepts of ontology is based on the calculation of the maximum flow between two given nodes of an oriented graph (Ford-Fulkerson method [30]).

Consider the ontology of computer science terms. The fragment of the corresponding owl file has the form shown in Fig. 3. Parsing the owl ontology file will reveal the connections between the concepts.

```
<!--
////////////////////////////////////////////////////////////////////////
//
// Individuals
//
////////////////////////////////////////////////////////////////////////
-->

<!-- urn:absolute:Ontology_02#Algorithm -->

<owl:NamedIndividual rdf:about="urn:absolute:Ontology_02#Algorithm">
    <rdf:type rdf:resource="urn:absolute:Ontology_02#Term"/>
    <UsedIn rdf:resource="urn:absolute:Ontology_02#Program"/>
    <definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        скінченна послідовність команд, виконання яких приводить до результату
    </definition>
    <keyword rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Алгоритм
    </keyword>
</owl:NamedIndividual>


<!-- urn:absolute:Ontology_02#Program -->

<owl:NamedIndividual rdf:about="urn:absolute:Ontology_02#Program">
    <rdf:type rdf:resource="urn:absolute:Ontology_02#Term"/>
    <UsesOf rdf:resource="urn:absolute:Ontology_02#Algorithm"/>
    <definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        алгоритм, записаний за допомогою мови програмування
    </definition>
    <keyword rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Програма
    </keyword>
</owl:NamedIndividual>
```

Figure 3: A fragment of an owl file that contains an ontology of computer science terms

The Computer Science Ontology is an explanatory dictionary that contains a set of terms, each term being a `<keyword, definition>` pair. The `Individuals` section of the ontology will be of interest to us in the first place because it contains the definition of terms. The definition of each concept term begins with the tag `owl:NamedIndividual`, its identifier is contained in the `rdf:about` attribute. A subsection that begins with the `keyword` tag contains the keyword of the term, and a subsection that begins with the `definition` tag contains its definition. Concepts related to the current term "uses-of" are listed in subsections that correspond to `UsesOf` tags, and their identifiers are the value of the `rdf:resource` attribute. The terms-concepts associated with the current used-in concept are listed in the subsections that correspond to the `UsedIn` tags, and their identifiers are also the value of the `rdf:resource` attribute. Thus, the "uses-of" relationship between the terms of the ontology can be schematically depicted as follows (Fig. 4):

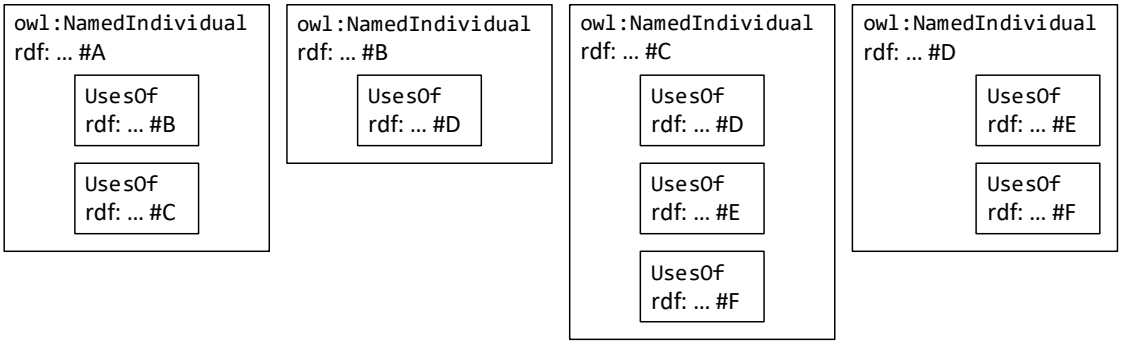

Figure 4: The "uses-of" relationships scheme between ontology terms

As you can see, the ontology owl file represents related terms as a list of contiguous vertices of the oriented graph (Fig. 5):

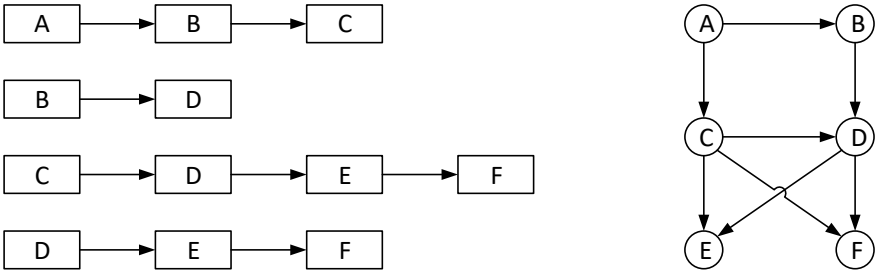

Figure 5: List of adjacent vertices and the corresponding oriented ontology graph

To calculate the distance between two terms-concepts of ontology, one must find all the ways from one concept to another. To do this, solve the traffic flow problem in an oriented ontology graph from the source node corresponding to the first concept to the receiving node corresponding to the second concept.

To simplify the calculations, discard all paths with a length of more than 4 transitions. Such paths will not be essential for calculating the semantic size of the text.

The algorithm is implemented by means of C#, Neo4j database is used to store intermediate results.

The SDIO (Semantic Distance In Ontology) function gets the ID of two terms:

```
public async Task<decimal> SDIO(long from, long to)
```

We need to get the number of paths from one term to another in an ontology with a certain number of transitions (from one to four):

```
for (int i = 0; i < 4; i++)
{
    var countResult = await session.RunAsync(
        _transactions.GetCountOfTransitions(from, to, i)
    );
    var record = await countResult.SingleAsync();
    var count = (long)(record.Values.Single().Value);
    dictionary.Add(i + 1, count);
}
```

We write down the results in the dictionary, with the key – the number of transitions.

After obtaining the number of paths between terms, calculate their distance in the ontology:

```
var N = 0m;
foreach (var item in dictionary.Keys)
{
      N += dictionary[item] / item;
}
var L = 1m / N;
```

`L` – the result of calculations, the result of the function.

For optimization, you can save the results of calculating the distance between a pair of terms, because the distance between the terms of the ontology does not change often (only when editing the ontology). At the beginning of the algorithm, we will check whether the algorithm has already been executed for this pair of terms, if so, we will return the result, without re-execution.

```
var sdio = SDIOTerms
    .Where(v => (v.From == from && v.To == to)
             || (v.From == to && v.To == from))
    .FirstOrDefault();

if(sdio != null)
{
    return sdio.L;
}
```

```
SDIOTerms.Add(new SDIOTerm
{
    L = L,
    To = to,
    From = from
});
```

4.2. Calculating the distance between terms in a scientific text

The calculation of the distance between two terms from the ontology in the scientific text will be based on the calculation of the sum of pairwise distances between each occurrence of these terms in the scientific text.

The SDIT (Semantic Distance In the Text) function gets two terms, between which you need to find the distance and structured text (an object in which paragraphs and sentences are clearly separated, as well as the terms used in it). Terms are passed using their ID.

```
public async Task<decimal> SDIT(long from, long to, StructuredText text)
```

Text structuring is implemented using regular expressions.

We obtain the semantic distance between these terms in the ontology:

```
var L = await SDIO(from, to);
```

We will record the results in a collection – a dictionary, the keys of which will be the number of transitions between terms from the text:

```
var Ns = new Dictionary<long, long>()
{
    { 0, 0 },
    { 2, 0 },
    { 4, 0 },
};
```

Implementation of the calculation of the number of transitions between terms in the text:

```
var textParagraphs = new List<StructuredParagraph>();
textParagraphs.AddRange(text.Paragraphs);

foreach (var paragraph in text.Paragraphs)
{
    var paragraphSentances = new List<StructuredSentence>();
    paragraphSentances.AddRange(paragraph.Sentences);

    foreach (var sentence in paragraph.Sentences)
    {
        if(sentence.Terms.Any(v => v.Id == from)
                    && sentence.Terms.Any(v => v.Id == to))
        {
            Ns[0] += sentence.Terms.Count(v => v.Id == from)
                    * sentence.Terms.Count(v => v.Id == to);
        }
        paragraphSentances.Remove(sentence);
        var sentencesTerms = paragraphSentances
                    .SelectMany(v => v.Terms);

        Ns[2] += sentence.Terms.Count(v => v.Id == from)
                    * sentencesTerms.Count(v => v.Id == to)
                    + sentence.Terms.Count(v => v.Id == to)
                    * sentencesTerms.Count(v => v.Id == from);
    }
    textParagraphs.Remove(paragraph);

    var paragraphsTerms = textParagraphs
             .SelectMany(v => v.Terms);
```

```
        Ns[4] += paragraph.Terms.Count(v => v.Id == from)
                 * paragraphsTerms.Count(v => v.Id == to)
                 + paragraph.Terms.Count(v => v.Id == to)
                 * paragraphsTerms.Count(v => v.Id == from);
    }
```

Now let us calculate the semantic distance:

```
    var Re = ((Ns[0] * 1m) / (L + 0m))
           + ((Ns[2] * 1m) / (L + 2m))
           + ((Ns[4] * 1m) / (L + 4m));
```

`Re` – the result of the function – the semantic distance between terms in the text.

4.3. Calculation of the semantic weight of a scientific text in relation to ontology

Semantic weight is the sum of the inverse distances between all ontology terms in a scientific text.

The SWOT (Semantic Weight Of the Text) function gets structured text:

```
    public async Task<decimal> SWOT(StructuredText text)
```

Implementation is quite simple, as all the necessary algorithms have already been implemented. All you need to do is find the semantic distances between each pair of terms in the text and sum up the inverse values.

First, we get all the terms:

```
    var terms = text.Terms.Distinct(new TermComparer()).ToList();
    var otherTerms = terms.ToList();

    public class TermComparer : IEqualityComparer<Term>
    {
        public bool Equals([AllowNull] Term x,
                 [AllowNull] Term y)
        {
            return x.Id == y.Id;
        }
        public int GetHashCode([DisallowNull] Term obj)
        {
            return obj.GetHashCode();
        }
    }
```

Implementation of the semantic weight calculation algorithm:

```
    var SW = 0m;

    foreach (var from in terms)
    {
        otherTerms.Remove(from);
        foreach (var to in otherTerms)
        {
            SW += 1m / (await SDIT(from.Id, to.Id, text));
        }
    }
    return SW;
```

4.4. Automatic abstracting of scientific texts

These algorithms can be used to solve the problem of automatic abstracting of texts. To do this, select fragments of text (sentences, paragraphs) with the greatest semantic weight.

5. Results

The result of the developed program is verified on the example of an ontology formed based on an explanatory dictionary of computer science [31]. Since the created program has a Ukrainian-language interface, the English translation of the inscriptions on the controls is provided with the help of notes.

The text from Wikipedia, the article "Computer programming" (Fig. 6) was used for testing:

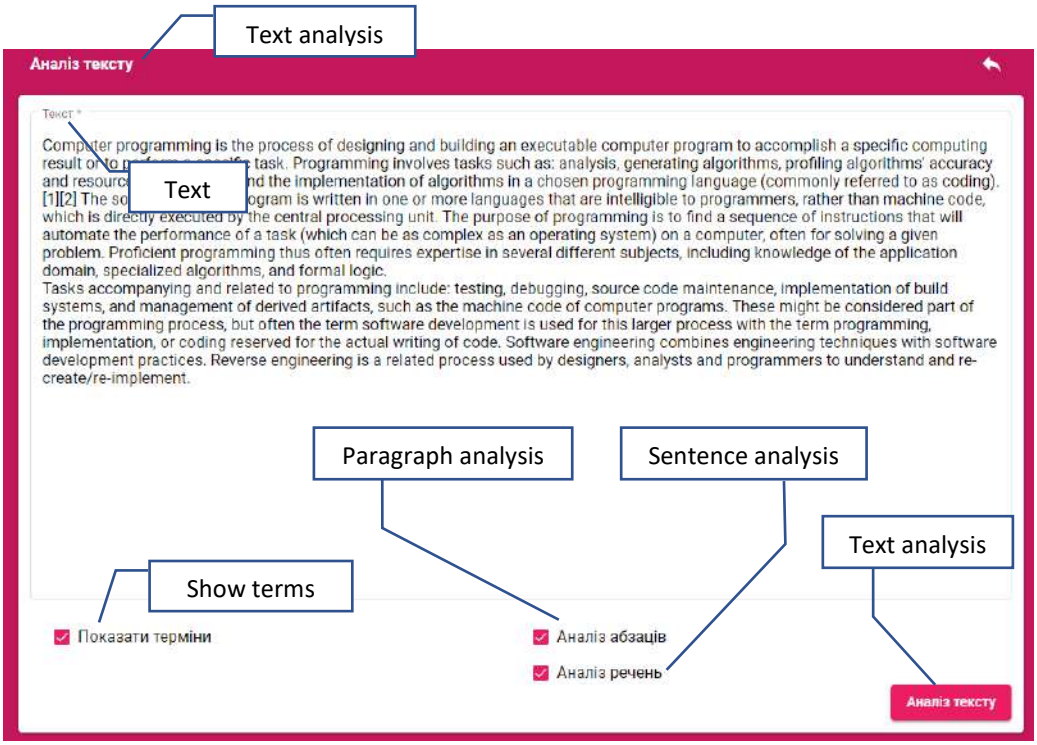

Figure 6: Form of text analysis

The result: 30394.102 is a very large number. This means that such a text really applies to the field of "Computer Science" (Fig. 7).

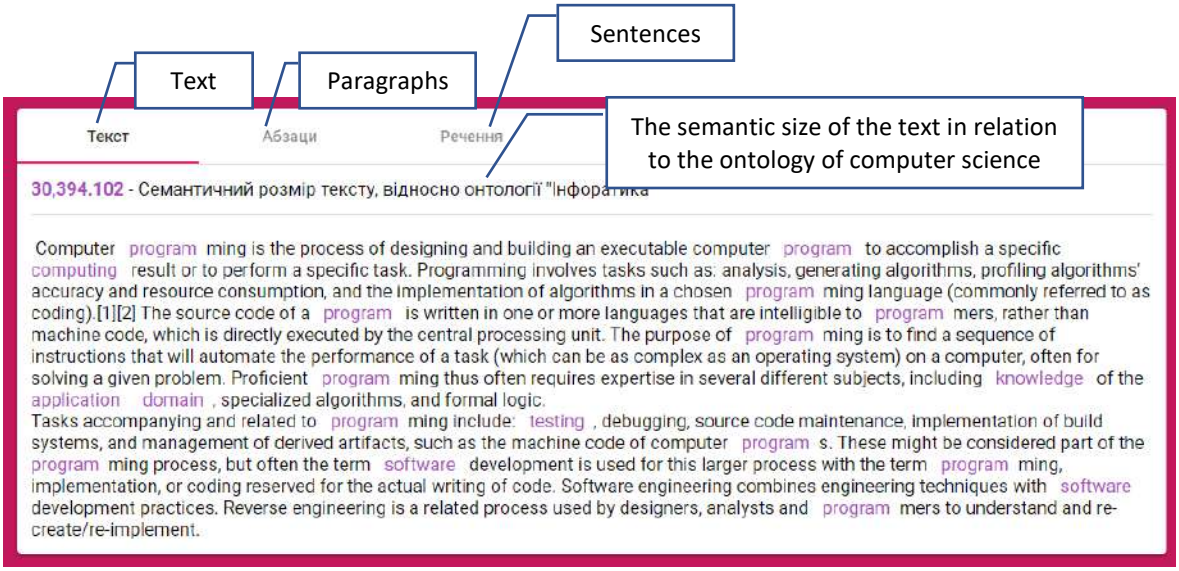

Figure 7: The result of text analysis

The results of the analysis of paragraphs and sentences are shown in Fig. 8 and Fig. 9.

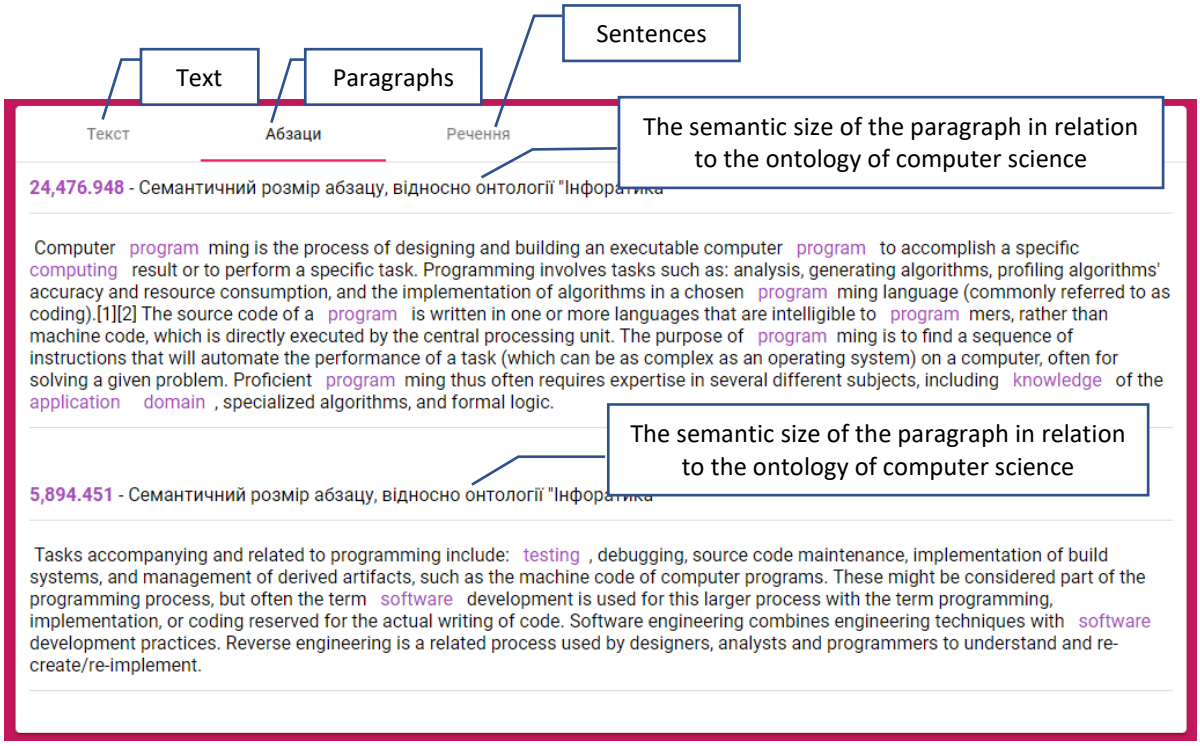

Figure 8: The result of paragraph analysis

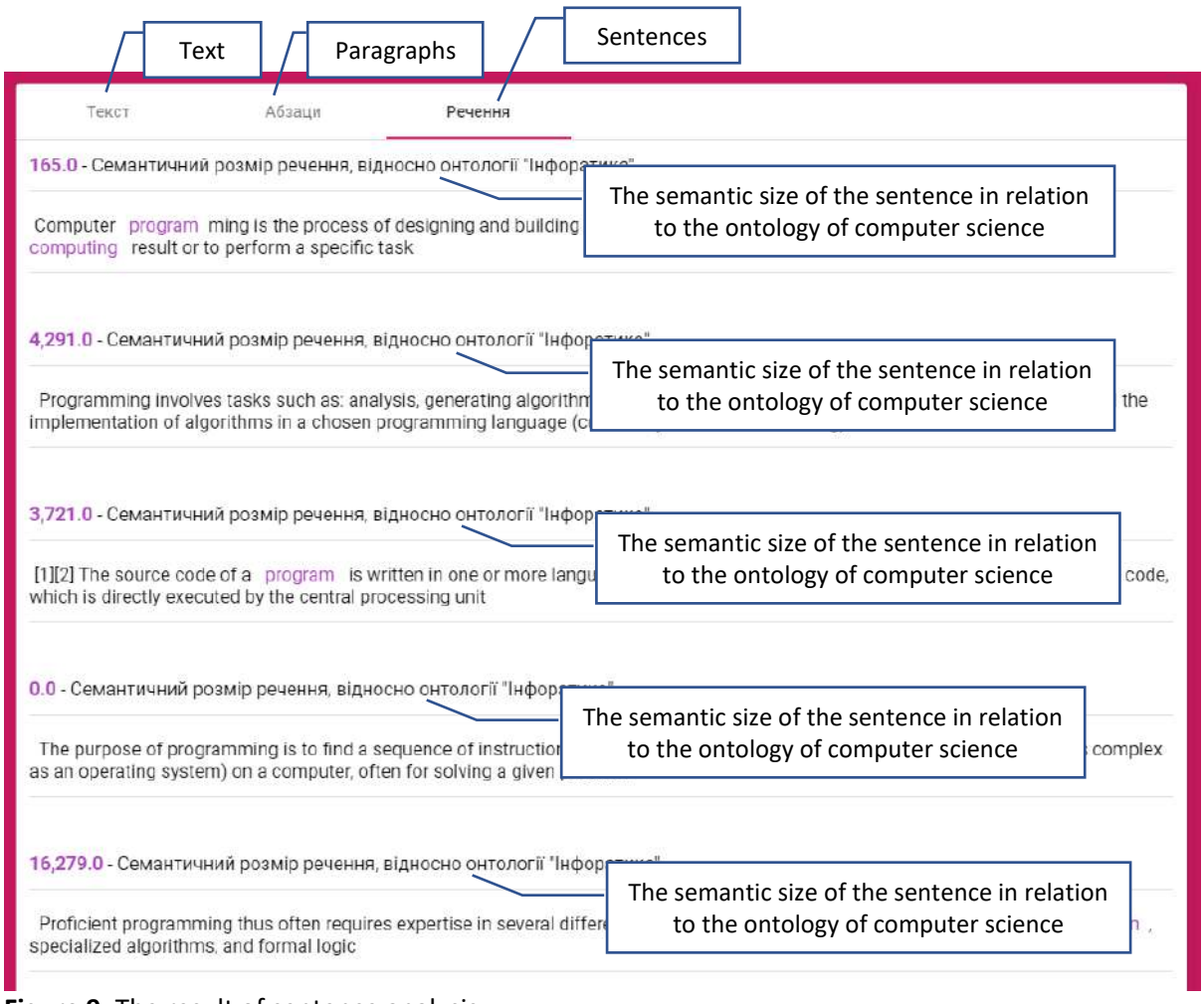

Figure 9: The result of sentence analysis

For comparison, we test a text that is not related to computer science – an article about coffee (Fig. 10).

The result is 1,999 – a very small number, which means that the analyzed text does not belong to the field of relevant ontology.

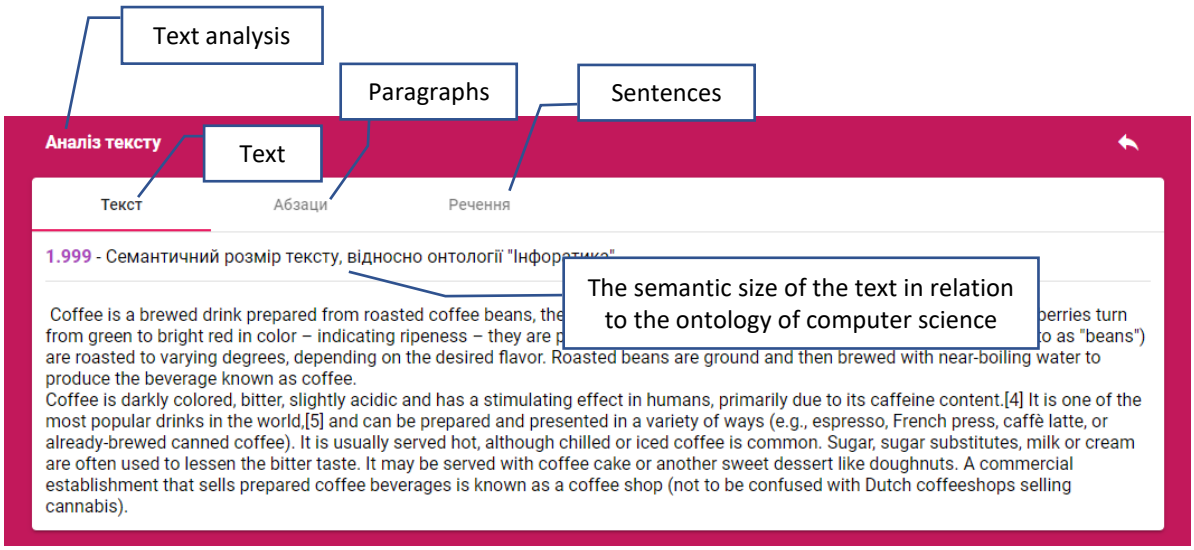

Figure 10: The result of text analysis from another field

6. Discussions

The implementation of these algorithms should take into account cases where the ontology graph contains "critical nodes", which are the intersections of different paths from one concept term to another.

6.1. Calculation of the distance between the concepts of ontology in the presence of critical nodes

Consider the case where there is a "crossroads" of paths leading from one concept of ontology to another – that is, a "crossroads" of paths between nodes of the oriented ontology graph.

Assume that the distance between nodes *A* and *E* along the path that passes through node *B* is N_1; the distance between *A* and *E* through nodes *C* and *D* is equal to N_2; the distance between *E* and *I* through nodes *F*, *G* is equal to N_3; the distance between *E* and *I* through *H* is N_4 (Fig. 11):

Here, node *E* is critical because it is the "crossroads" of the *A-B-E-H-I* and *A-C-D-E-F-G-I* paths.

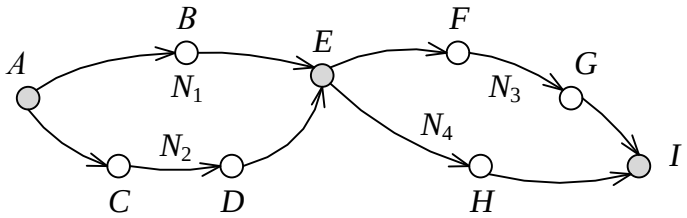

Figure 11: Node *E* is critical on the path from *A* to *I*

$$\begin{aligned} R(A_B E) &= N_1, \\ R(A_{CD} E) &= N_2, \\ R(E_{FG} I) &= N_3, \\ R(E_H I) &= N_4 \end{aligned} \tag{8}$$

Calculate the distance between nodes *A* and *I* in two ways.

The first way – take into account the critical node-"crossroads" *E*:

$$R(A,I) = R(A,E) + R(E,I),$$
$$\frac{1}{R(A,E)} = \frac{1}{N_1} + \frac{1}{N_2} = \frac{N_1 + N_2}{N_1 N_2},$$
$$\frac{1}{R(E,I)} = \frac{1}{N_3} + \frac{1}{N_4} = \frac{N_3 + N_4}{N_3 N_4}, \quad (9)$$
$$R(A,I) = \frac{N_1 N_2}{N_1 + N_2} + \frac{N_3 N_4}{N_3 + N_4},$$
$$R(A,I) = \frac{N_1 N_2 (N_3 + N_4) + (N_1 + N_2) N_3 N_4}{(N_1 + N_2)(N_3 + N_4)}$$

The second way is to consider each path independently of the others, ignoring the "crossroads":

$$\frac{1}{R(A,I)} = \frac{1}{R(A_{BEFG}I)} + \frac{1}{R(A_{BEH}I)} + \frac{1}{R(A_{CDEFG}I)} + \frac{1}{R(A_{CDEH}I)},$$
$$\frac{1}{R(A,I)} = \frac{1}{N_1 + N_3} + \frac{1}{N_1 + N_4} + \frac{1}{N_2 + N_3} + \frac{1}{N_2 + N_4} \quad (10)$$

Consider the case when N_1=1, N_2=2, N_3=2, N_4=1.

According to the formula (9), $R(A, I) = R(A, E) + R(E, I)$; $1/R(A, E) = 1 + 1/2 = 3/2$; $R(A, E) = 2/3 = R(E, I)$. From here $R(A, I) = 4/3$.

According to the formula (10), $1/R(A, I) = 1/2 + 2/3 + 1/4 = 6/12 + 8/12 + 3/12 = 17/12$; $R(A, I) = 12/17 < 4/3$ – the values calculated by formulas (9) and (10) differ.

For the case shown in Fig. 3, N_1=2, N_2=3, N_3=3, N_4=4.

According to the formula (9), $1/R(A, E) = 1/2 + 1/3 = 5/6$; $R(A, E) = 6/5 = R(E, I)$. From here $R(A, I) = 12/5 = 60/25$.

According to the formula (10), $1/R(A, I) = 1/4 + 2/5 + 1/6 = 15/60 + 24/60 + 10/60 = 49/60$, $R(A, I) = 60/49 < 60/25$ – the values do not match again.

Thus, the presence of critical nodes-"crossroads" does not allow the use of formula (1) to calculate the distance between the concepts of ontology in a way that ignores such critical nodes. Therefore, the algorithm for detecting nodes-"crossroads" will be significant.

6.2. Detection of critical nodes

A critical node is a node that corresponds to a single minimum section of the graph, i.e. the smallest section is equal to 1. The problem of finding the smallest section of the graph is twofold to the problem of the largest flow [30].

7. Conclusions

This paper proposes a method of semantic analysis based on inverse-additive metrics, which takes into account the semantic distance between the terms of the ontology in the text document being analyzed. This metric allows you to correctly process cases where there are several paths in the oriented graph of the ontology from one concept node to another.

Semantic analysis of scientific documents is considered, as such texts have a clear structure. The concept of semantic distance between the terms of a scientific text and the semantic weight of a scientific text is introduced. The semantic weight of individual fragments of a scientific text can be used to solve the problem of automatic abstracting.

Some of the difficulties in implementing the proposed approach related to critical nodes on the path in the oriented ontology graph from one concept node to another are discussed.

8. References

[1] M. Saidova, Semantic Analysis of Literary Terms by Literary Types in “The Concise Oxford Dictionary of Literature Terms”, Philology Matters: Vol. 2021, Iss. 1, Article 11 (2021) 118-138. doi:10.36078/987654486.

[2] N. Panasenko, Semantic Structure of Literary Text, Zeszyty Naukowe Uniwersytetu Rzeszowskiego. Seria Filologiczna. Studia Anglica Resoviensia 10 (2021) 38-50.
[3] Susan T. Dumais, Latent Semantic Analysis, Annual Review of Information Science and Technology 38 (2005) 188–230. doi:10.1002/aris.1440380105.
[4] P. Foltz, Latent Semantic Analysis for Text-Based Research. Behavior Research Methods, Instruments, & Computers, 28 (2) (1996) 197-202. doi:10.3758/BF03204765.
[5] H. Raza, M. Faizan, A. Hamza, A. Mushtaq, N. Akhtar, "Scientific Text Sentiment Analysis using Machine Learning Techniques", International Journal of Advanced Computer Science and Applications (IJACSA) 10 (12) (2019). URL: http://dx.doi.org/10.14569/IJACSA.2019.0101222.
[6] L. Ogiela, Intelligent Cognitive Information Systems in Management Applications, Cognitive Information Systems in Management Sciences (2017) 79-122. doi:10.1016/B978-0-12-803803-1.00006-9.
[7] N. Gupta, R. Agrawal, Application and Techniques of Opinion Mining, Hybrid Computational Intelligence (2020) 1-23. doi:10.1016/B978-0-12-818699-2.00001-9.
[8] W. Hodges, Functional Modelling and Mathematical Models: A Semantic Analysis, Philosophy of Technology and Engineering Sciences (2009) 665-692. doi:10.1016/B978-0-444-51667-1.50029-X.
[9] L.Ogiela, M.R. Ogiela, Cognitive Information Systems, Advances in Cognitive Information Systems. Cognitive Systems Monographs, vol 17, Springer, Berlin, Heidelberg, 2012, pp. 51–60. URL: https://doi.org/10.1007/978-3-642-25246-4_3.
[10] O. Levchenko, O. Tyshchenko, M. Dilai, Automated Identification of Metaphors in Annotated Corpus (Based on Substance Terms), in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 16-31.
[11] H. Schöpper, W. Kersten, Using Natural Language Processing for Supply Chain Mapping: A Systematic Review of Current Approaches, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 71-86.
[12] N. Kunanets, Y. Oliinyk, D. Myhal, K. Shunevych, A. Rzheuskyi, Y. Shcherbyna, Enhanced LSA Method with Ukraine Language Support, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 129-140.
[13] K. Jindal, R Aron, A systematic study of sentiment analysis for social media data, Materials Today (2021). URL: https://www.sciencedirect.com/science/article/pii/S2214785321000705.
[14] B. Ozyurt, M. Ali Akcayol, A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA, Expert Systems with Applications. URL: https://www.sciencedirect.com/science/article/pii/S0957417420309519.
[15] V. Shyrokov, "Accuracy" vs "Unambiguity" in Linguistics, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1-5.
[16] L. Savytska, N. Vnukova, I. Bezugla, V. Pyvovarov, M. Turgut Sübay, Using Word2vec Technique to Determine Semantic and Morphologic Similarity in Embedded Words of the Ukrainian Language, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 235-248.
[17] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, in: Proceedings of Workshop at ICLR 2013, Computation and Language Scottsdale, Arizona, USA, 2013. arXiv:1301.3781v3.
[18] R. Lebret, R. Collobert, Word Embeddings through Hellinger PCA, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 482–490. doi:10.3115/v1/E14-1051.
[19] V. Vysotska, S. Holoshchuk, R. Holoshchuk, A Comparative Analysis for English and Ukrainian Texts Processing Based on Semantics and Syntax Approach, in: Proceedings of the 5th

International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 311-356.
[20] E. Halling, J. Vain, A. Boyarchuk, O. Illiashenko, Test Scenario Specification Language for Model-Based Testing, International Journal of Computing, volume 18(4) (2019) 408-421.
[21] T. Basyuk, A. Vasyliuk, Approach to a Subject Area Ontology Visualization System Creating, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 528-540.
[22] A. Vasyliuk, T. Basyuk, Construction Features of the Industrial Environment Control System, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1011-1025.
[23] S. Orekhov, H. Malyhon, N. Stratienko, T. Goncharenko, Software Development for Semantic Kernel Forming, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1312-1322.
[24] J. Vishal, S. Mayank, Ontology based information retrieval in semantic web: a survey, International Journal of Information Technology and Computer Science, Vol.5, No.10 (2013) 62-69. doi: 10.5815/ijitcs.2013.10.06.
[25] O. Pashchetnyk, V. Lytvyn, V. Zhyvchuk, L. Polishchuk, V. Vysotska, Z. Rybchak, Y. Pukach, The Ontological Decision Support System Composition and Structure Determination for Commanders of Land Forces Formations and Units in Ukrainian Armed Forces, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1077-1086.
[26] O. Duda, V. Pasichnyk, H. Lypak, N. Veretennikova, N. Kunanets, O. Matsiuk, V. Mudrokha, Formation of Integrated Repositories of Social and Communication Data by Consolidating the Resources of Museums, Libraries and Archives in Smart Cities Projects, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1420-1430.
[27] L. Chyrun, V. Andrunyk, L. Chyrun, A. Gozhyj, A. Vysotskyi, O. Tereshchuk, N. Shykh, V. Schuchmann, The Electronic Digests Formation and Categorization for Textual Commercial Content, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 1816-1831.
[28] A. Berko, V. Andrunyk, L. Chyrun, M. Sorokovskyy, O. Oborska, O. Oryshchyn, M. Luchkevych, O. Brodovska, The Content Analysis Method for the Information Resources Formation in Electronic Content Commerce Systems, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021. – P. 1632-1651.
[29] Hryhorovych V, Construction of semantic metric for measuring the distance between ontology concepts, in: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Vol. I: main conference, Kharkiv, Ukraine, April 22-23, 2021, pp. 498–510.
[30] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 4th. ed., MIT Press Academic, 2022.
[31] H. Korotenko, L. Korotenko, H. Pivniak, Tlumachnyi slovnyk z informatyky [Explanatory dictionary of computer science], Dnipropetrovs'k, 2010.