Calculation of the Semantic Distance between Ontology Concepts: Taking into Account Critical Nodes

Viktor Hryhorovych

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

Abstract

The problem of semantic analysis of textual information does not lose its relevance. It is reduced to a quantitative assessment of the elements of a text document and the relationships between them.

This work develops a method of semantic analysis based on the inverse-additive metric, which takes into account the semantic distance between terms from the ontology. This metric allows you to correctly process cases when there are several paths in the directed graph of the ontology from one concept node to another.

The work describes ways to overcome some difficulties in the implementation of this method, associated with critical nodes on the path in the directed graph of the ontology from one concept node to another. Critical nodes are intersection nodes, merge nodes and branch nodes. The presence of such nodes requires additional analysis of the ontology graph and additional calculations to correctly calculate the semantic distance between ontology concepts.

Keywords 1

semantic analysis, semantic metric, ontology, semantic distance, critical nodes

1. Introduction

Tasks related to the semantic analysis of textual information do not lose their relevance. These tasks include the tasks of automating the filtering, classification, and clustering of text documents, automating the abstracting of a given text, automating the evaluation of answers to open test tasks, automatic construction of a semantic network for a given text, etc.

To solve such problems, we should perform a quantitative assessment of the elements of the text document and the relationships between them.

Many approaches are based on quantitative characteristics of text documents, but most are actually parsing. In particular, these approaches use the frequency of occurrences of keywords. Other methods are based on the transformation of texts into vectors of real numbers and their use for quantitative comparison of corresponding text documents. Such approaches also use calculating the number of occurrences of certain keywords or comparing with some template - the basic body of text information.

This work continues to develop a semantic analysis method based on an inverse-additive metric that takes into account the semantic distance between ontology terms in a text document. Such a metric allows us to correctly process cases when there are several paths in the directed graph of the ontology from one concept node to another.

When implementing this method, one should overcome some difficulties associated with critical nodes on the path in the directed ontology graph from one concept node to another. Critical nodes are intersection nodes, merge nodes and branch nodes. The presence of such nodes requires additional analysis of the ontology graph and additional calculations to correctly calculate the semantic distance between ontology concepts.

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: viktor.grigorovich@gmail.com; viktor.h.hryhorovych@lpnu.ua (V. Hryhorovych)
ORCID: 0000-0002-5828-067X (V. Hryhorovych)

2. Related Works

Semantic analysis of texts does not lose its relevance; many works in the field of information technologies and computational linguistics are devoted to developing and applying information technologies for the semantic analysis of texts.

Work [1] describes a method based on the fact that words close in meaning occur in similar fragments of the text. A matrix is constructed containing the number of words per document (rows represent unique words and columns represent each document). Documents are compared based on the scalar product between the normalizations of the two vectors formed by the corresponding two columns. Values close to 1 represent semantically close documents, while values close to 0 represent very different documents. Work [2] is the result of a study of scientific texts based on a previously constructed corpus of citations. The work [3] is devoted to the description of intelligent information systems for semantic analysis, semantic interpretation, and understanding of data designed to support data management processes. The work [4] describes the method of semantic analysis based on the semantic orientation of vocabulary. The work [5] describes the application of multiple theoretical approaches to semantic analysis. The work [6] is devoted to the application of the method of latent semantic analysis (see [7] and [8]) for the processing of Ukrainian-language texts. Work [9] describes the method of converting words into vectors of real numbers. The application to the Ukrainian corpus is interesting since word vectors are formed and processed according to the rules of the Ukrainian language. The work [10] describes the application of the method of generative grammar in linguistic modeling. The work [11] describes the results of forming a semantic core for a web resource. Work [12] is devoted to the problem of choosing a document representation suitable for creating user profiles and for supporting the content-based search process.

All these and many other works do not take into account the possibility of the existence of many paths from one ontology term to another.

3. Methods

3.1. Inverse-additive metric for ontology concepts

The inverse-additive metric introduced in [13] allows calculating the distance between concepts of the ontology in the case when there are several paths from one concept to another. Consider a directed graph of the ontology, which represents concepts and connections between them: each concept corresponds to a node, and to each connection – an edge of the directed graph. If the ontology represents an explanatory dictionary, then each term is a pair consisting of a keyword and its interpretation; and the text of the interpretation contains other keywords – references to other terms. An edge of a directed graph connects the node corresponding to the keyword of the corresponding term with the keyword of another term used in the interpretation of the first term. Since the interpretation of different terms can use the same keywords, this is the reason for the existence of several paths from one node to another in the directed ontology graph.

By analogy with electrical resistance in parallel and series connections, the inverse-additive metric determines the distance $R(A, B)$ between concepts A and B as follows:

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

where

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

K – is the number of different paths that can be taken along the directed graph of a certain 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:

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

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

The given definition satisfies all metric axioms. It should be noted that for the axiom of symmetry, a pair of complementary symmetric relations should be introduced. For example, for an ontology-explanatory dictionary, this is a pair of connections "uses-of" – "used-in", which allows ensuring the fulfillment of the axiom of symmetry for the proposed metric in the following interpretation:

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

3.2. Critical nodes of the ontology graph

Cases should be considered separately when the ontology graph contains "critical nodes", which are the intersection, merging, or branching of different paths from one term concept to another.

3.2.1. The distance between ontology concepts in the presence of intersection nodes, merging nodes, and branching nodes

The paper [14] describes the case when there is an "intersection" of paths leading from one ontology concept to another, i.e., an "intersection" of paths between the nodes of the directed graph of the ontology.

Consider the case depicted in Fig. 1.

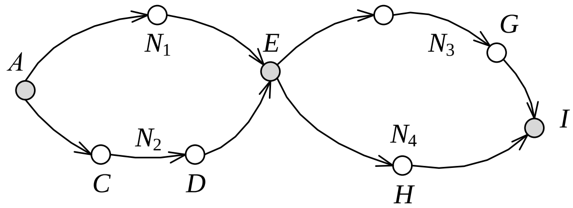

Figure 1: Node *E* is an intersection that is critical on the way from *A* to *I*

We denote the distance between nodes *A* and *E* along the path that passes through node *B* is equal to 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 equal to N_4:

$$\begin{gathered} R(A_B E) = N_1, \\ R(A_{CD} E) = N_2, \\ R(E_{FG} I) = N_3, \\ R(E_H I) = N_4 \end{gathered} \tag{4}$$

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

Then:

$$\begin{gathered} R(A,I) = R(A,E) + R(E,I), \\ \frac{1}{R(A,E)} = \frac{1}{N_1} + \frac{1}{N_2} = \frac{N_1 + N_2}{N_1 N_2}, \\ \frac{1}{R(E,I)} = \frac{1}{N_3} + \frac{1}{N_4} = \frac{N_3 + N_4}{N_3 N_4}, \\ R(A,I) = \frac{N_1 N_2}{N_1 + N_2} + \frac{N_3 N_4}{N_3 + N_4} \end{gathered} \tag{5}$$

For the case shown in Fig. 1, N_1=2, N_2=3, N_3=3, N_4=2.

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

When directly calculating the semantic distance, ignoring critical nodes and considering each path independently of the others, we get:

$$\frac{1}{R(A,I)} = \frac{1}{R(A_{BEFG}I)} + \frac{1}{R(A_{BEH}I)} + \frac{1}{R(A_{CDEFG}I)} + \frac{1}{R(A_{CDEH}I)},$$
$$\frac{1}{R(A,I)} = \frac{1}{N_1 + N_3} + \frac{1}{N_1 + N_4} + \frac{1}{N_2 + N_3} + \frac{1}{N_2 + N_4} \tag{6}$$

According to the formula (6), $1/R(A, I) = 1/4 + 2/5 + 1/6 = 15/60 + 24/60 + 10/60 = 49/60$, $R(A, I) = 60/49 < 60/25$ – the semantic distance value will be false.

Therefore, the presence of critical "intersection" nodes does not allow directly using formulas (1-2) to calculate the distance between concepts of the ontology, ignoring the existence of critical nodes. Therefore, it will be essential to identify "intersection" nodes.

Similar problems arise in the presence of merging nodes (Fig. 2) and branching nodes (Fig. 3).

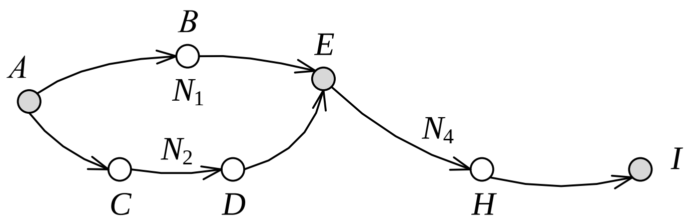

Figure 2: Node *E* is a merge that is critical on the path from *A* to *I*

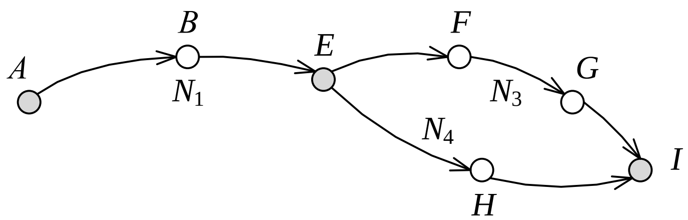

Figure 3: Node *E* is a branch that is critical on the path from *A* to *I*

If there is a merging node (Fig. 2), we will get:

$$R(A,I) = R(A,E) + R(E,I),$$
$$\frac{1}{R(A,E)} = \frac{1}{N_1} + \frac{1}{N_2} = \frac{N_1 + N_2}{N_1 N_2},$$
$$R(E,I) = N_4, \tag{7}$$
$$R(A,I) = \frac{N_1 N_2}{N_1 + N_2} + N_4$$

For the case when (Fig. 2), $N_1=2$, $N_2=3$, $N_4=2$, this will lead to $1/R(A, E) = 1/2 + 1/3 = 5/6$; $R(A, E) = 6/5$; $R(E, I) = 2$. Hence $R(A, I) = 16/5$.

Direct application of formulas (1-2) without taking into account the presence of a critical node will give

$$\frac{1}{R(A,I)} = \frac{1}{R(A_{BEH}I)} + \frac{1}{R(A_{CDEH}I)}, \tag{8}$$

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

or $1/R(A, I) = 1/4 + 1/5 = 9/20$, hence $R(A, I) = 20/9$ – again a false value.

Similarly, it will be for the branching node (Fig. 3):

$$R(A,I) = R(A,E) + R(E,I),$$

$$R(A,E) = N_1,$$

$$\frac{1}{R(E,I)} = \frac{1}{N_3} + \frac{1}{N_4} = \frac{N_3 + N_4}{N_3 N_4}, \tag{9}$$

$$R(A,I) = \frac{N_3 N_4}{N_3 + N_4} + N_1$$

For the case when (Fig. 3), N_1=2, N_3=3, N_4=2, this will lead to $R(A, E) = 2$; $1/R(E, I) = 1/2 + 1/3 = 5/6$; $R(E, I) = 6/5$. Hence, as in the previous example, $R(A, I) = 16/5$.

Again, direct application of formulas (1-2) without taking into account the presence of a critical node will give

$$\frac{1}{R(A,I)} = \frac{1}{R(A_{BEH}I)} + \frac{1}{R(A_{BEFH}I)},$$

$$\frac{1}{R(A,I)} = \frac{1}{N_1 + N_3} + \frac{1}{N_1 + N_4} \tag{10}$$

or $1/R(A, I) = 1/4 + 1/5 = 9/20$, hence $R(A, I) = 20/9$ – we got a false value again.

Therefore, in order to obtain the correct value of the semantic distance between ontology concepts, it is important, first, to identify critical nodes – intersection nodes, merging nodes, and branching nodes; secondly, to use the correct algorithms for calculating the semantic distance in the presence of critical nodes.

3.2.2. Presentation of concepts of the ontology of terms in the form of a list of adjacent vertices and identification of critical nodes

As stated in [14], the directed graph of the ontology of terms can be represented as a list of adjacent vertices.

Then the list of adjacent vertices for the case of the intersection node from Fig. 1 will be like this (Fig. 4):

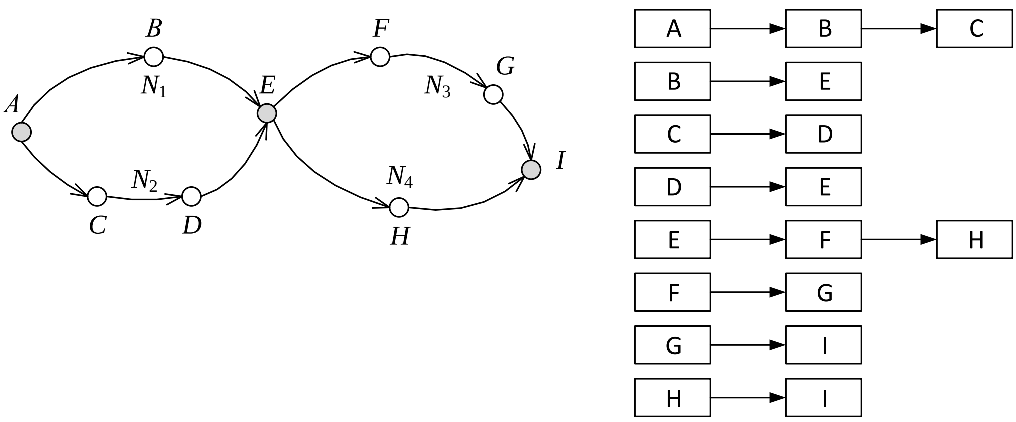

Figure 4: List of adjacent vertices for the intersection node case

For the cases of a merging node (Fig. 2) and a branching node (Fig. 3), the lists of adjacent vertices will be as follows (Fig. 5 and Fig. 6, respectively):

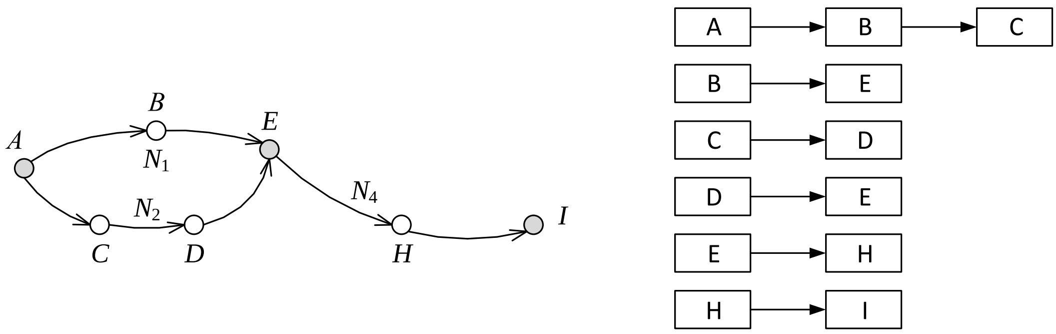

Figure 5: List of adjacent vertices for the merge node case

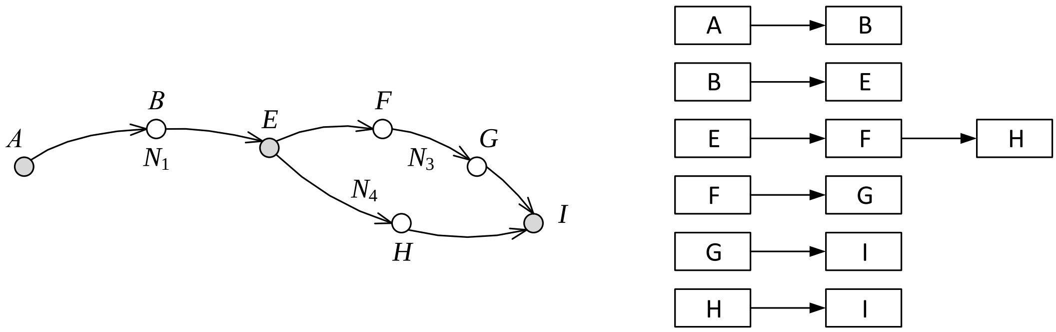

Figure 6: List of adjacent vertices for the branch node case

It is easy to see that in the case of a merge node, the critical node is an intermediate node that acts as a receiver more than once: $B \rightarrow E$, $D \rightarrow E$.

For the case of a branching node, the intermediate node that is the source more than once will be critical: $E \rightarrow F \rightarrow H$.

For an intersection node, both conditions are true: such a critical node is an intermediate node that simultaneously acts as a receiver more than once and is a source more than once.

Therefore, to identify critical nodes, it is enough to look at the list of adjacent vertices from the initial node (A) to the final node (I) and find out whether there are intermediate nodes that are sources and/or receivers more than once.

3.3. Calculation of the semantic distance in the presence of critical nodes

In the presence of critical nodes, the direct application of formulas (1-2) to calculate the semantic distance between concepts-terms of the ontology leads to an incorrect result.

In this case, one should use the methods of calculating the equivalent resistance with a series-parallel connection of resistors [15-19].

Such methods of calculating complex electric circuits are Nodal analysis, Mesh analysis, Superposition, Effective medium approximations, method of nodal and contour equations; method of contour currents; overlay method; nodal voltage method; equivalent generator method, etc.

Brief characteristics of some of these methods:

Nodal analysis method: The number of unknown variables (voltage variables) and equations to be solved is equal to the number of nodes minus one. Each voltage source connected to the reference node reduces the number of unknown variables and equations by one. This method is very useful when the circuit has voltage sources. The nodal analysis uses the concept of nodal voltage and treats the nodal voltages as unknown variables. For all nodes except the selected reference node, the node voltage is defined as the voltage drop from the node to the reference node. Thus, there are N-1 node voltages for a circuit with N nodes. The nodal analysis uses Kirchhoff's current laws (KCL) at N-1 nodes to obtain N-1 independent equations. [17] Since the equations generated by KCL involve currents entering and leaving nodes, these currents, if their values are unknown, must be represented by unknown variables (node voltages). For some elements (such as resistors and capacitors), deriving the element currents from the node voltage is trivial. For some common elements, where this is not possible, special methods are developed. For example, a concept called a supernode is used for circuits with independent voltage sources. Algorithm of the node analysis method: Mark all the nodes of the chain. Arbitrarily select any node as a reference node. Determine the voltage variable from each remaining node to the reference. These alternating voltages must be defined as the voltage rise relative to the reference node. Write the KCL equation for each node except the support node. Solve the resulting system of equations.

Mesh analysis: the number of unknown variables (current variables) and equations to be solved is equal to the number of loops. Each current source connected to the circuit reduces the number of unknown variables by one. However, this method can be used only when the investigated circle can be represented as a flat network so that no branch intersects with another. This method is very good when there are current sources in the circuit.

A loop is a loop that does not contain an inner loop. Count the number of contours in the diagram. Assign a grid current to each window pane. Write the KVL equation for each grid whose current is unknown. Solve the resulting equations.

Mesh analysis (or loop current method) is a method used to solve planar circuits for currents (and indirectly voltages) anywhere in an electrical circuit. Planar circuits are circuits that can be drawn on a flat surface without crossing wires. A more general method called loop analysis (with corresponding network variables called loop currents) can be applied to any circuit, planar or not. Mesh analysis and loop analysis use Kirchhoff's voltage law to obtain a set of equations that are guaranteed to be solved if the circuit has a solution [20]. Mesh analysis is usually easier to use when the circuit is flat compared to loop analysis [21].

Method of nodal and contour equations. Kirchhoff's first and second laws apply to direct current electrical circuits. Rules for using the method:

1. Determine the total number of equations (equal to the number of unknown currents, that is, the number of branches).
2. Choose arbitrarily the direction of the currents. Currents directed to the node are considered positive, from the node – as negative.
3. Compose nodal equations (according to Kirchhoff's first law for a direct current circuit). The number of equations is equal to the number of nodes minus one.
4. Draw contour equations (according to Kirchhoff's second law for a direct current circuit). The number of contour equations is equal to the difference between the total number of equations and the number of nodal equations.
5. Solve the resulting system of equations using mathematical methods.

At the same time, the simplest circuits should be chosen, that is, circuits with a smaller number of EMF sources and resistors, and in each new circuit, there should be at least one line that is not included in the circuits for which the equations have already been compiled. Bypassing the contour is chosen arbitrarily. EMFs whose direction coincides with the direction of the circuit bypass are considered positive. The voltage drop across the resistor is positive where the current direction coincides with the bypass direction. Since the direction of the current is chosen arbitrarily, if we get a negative value during the calculation, then in reality the current flows in the opposite direction.

Method of contour currents. The method of nodal and contour equations is quite cumbersome (the greater the number of branches, the greater the number of equations in the system). The method of contour currents greatly simplifies the calculation, because it reduces the number of equations in the system. According to this method, equations are formed only according to Kirchhoff's second law for a direct current electric circuit, for which the required number of circuits is selected.

Superposition is probably the most conceptually simple method but quickly leads to a large number of equations and messy impedance combinations as the network becomes larger.

4. Experiment

4.1. Test program for finding critical nodes

The algorithm of the test program for finding critical nodes of a directed graph is as follows:

1. First, a directed graph is randomly generated, which is represented by an adjacency matrix.
2. Based on the adjacency matrix, a list of adjacent vertices is formed.
3. Critical nodes are searched for in the list of adjacent vertices. For this purpose, the width search algorithm is used.

4.2. Formation of a directed graph given by the adjacency matrix

To verify the correctness and effectiveness of the developed algorithms, test data should be generated. These inputs are directed graphs generated using a random number generator. These graphs will model ontologies of terms, vertices correspond to concepts, and edges correspond to connections between concepts.

A directed graph, which corresponds to the relationships between the concepts of the term ontology, should not have loops and cycles. The adjacency matrix of such a graph should be sufficiently sparse – the number of ones should be significantly less than the number of zeros.

To form a sufficiently sparse adjacency matrix, in which the number of ones is much smaller than the number of zeros, we use the following approach:

```
int n = 3;
for (int i = 0; i < size; i++)
	for (int j = 0; j < size; j++)
		m[i][j] = rand() % n == 0 ? 1 : 0;
```

The elements of the matrix randomly get the value 1 or 0 with probabilities 1/n and (n-1)/n, respectively. The value n = 3 was chosen for testing.

In order for the graph not to contain loops, the main diagonal of the adjacency matrix must contain only zeros:

```
for (int i = 0; i < size; i++)
	m[i][i] = 0;
```

Finally, to eliminate cycles consisting of two vertices, consider the elements located above the main diagonal. If the element m[i][j] has the value 1, then the value 0 will be assigned to the element m[j][i]:

```
for (int i = 0; i < size; i++)
	for (int j = i + 1; j < size; j++)
		if (m[i][j])
			m[j][i] = 0;
```

Cycles that pass through a larger number of vertices are much more difficult to detect and eliminate. Therefore, this test program does not perform such operations.

4.3. Formation and processing of the list of adjacent vertices

The algorithm for forming the list of adjacent vertices from the adjacency matrix is well-known and does not require comments.

Processing the list consists of finding all paths from one given node to another and checking whether these paths share any nodes other than the start and end nodes.

To search for these paths, the width search method was used, which also needs no explanation.

5. Results

A test program was created for finding critical nodes of a directed graph that models the ontology of terms: vertices correspond to concepts, and edges correspond to connections between concepts.

The results of the program are shown in Fig. 7:

```
Microsoft Visual Studio Debug Console
size = ? 8
  0  0  0  0  1  1  1  1
  0  0  0  0  1  0  0  0
  0  1  0  0  1  0  0  0
  0  0  1  0  0  0  1  0
  0  0  0  1  0  0  0  0
  0  1  0  1  0  0  1  1
  0  1  0  0  1  0  0  0
  0  0  0  0  0  0  0  0

Vertex #0 ::   4 -> 5 -> 6 -> 7
Vertex #1 ::   4
Vertex #2 ::   1 -> 4
Vertex #3 ::   2 -> 6
Vertex #4 ::   3
Vertex #5 ::   1 -> 3 -> 6 -> 7
Vertex #6 ::   1 -> 4
Vertex #7 ::

src  = ? 0
dest = ? 4

Vertex #0 ::   4 ->
Vertex #5 ::   1 ->
Vertex #3 ::   2 ->
Vertex #6 ::   1 ->
Vertex #4 ::   3
4
6
3 ->
```

Figure 7: Results of the test program

The performance of this test program has also been tested against predefined directed graphs when the input is a text file containing a list of adjacent vertices.

The results of the test examples confirm the correctness of the developed algorithms.

6. Discussions

The detection of critical nodes in itself requires additional time compared to the direct application of formulas (1-2).

If the check confirms the presence of critical nodes, then the correct calculation of the semantic distance leads to the need to solve the system of equations according to the chosen method (for example, methods of contour or nodal equations). Naturally, the question arises - is it worth doing all this, or would it be better to use some other less accurate but faster methods?

First of all, it should be noted that the proposed inverse-additive metric has its own field of application, namely term ontologies. Each term in such an ontology consists of a keyword and its definition.

Suppose keyword definition keyword_1 uses keywords keyword_2 and keyword_N, keyword definition keyword_2 uses keywords keyword_3 and keyword_K, keyword definition keyword_3 uses keywords keyword_K and keyword_N, keyword definition keyword_K uses keyword keyword_N, keyword definition keyword_N does not use keywords of this ontology of terms.

Then the given example of the ontology of terms can be schematically represented by the diagram shown in Fig. 8. The corresponding directed graph is shown in Fig. 9.

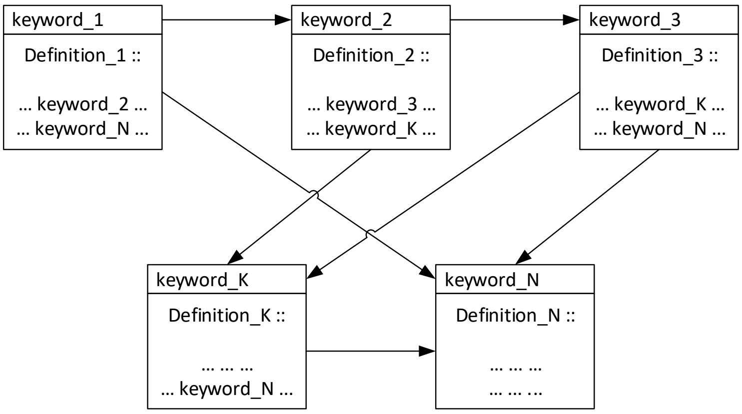

Figure 8: Diagram of the ontology of terms

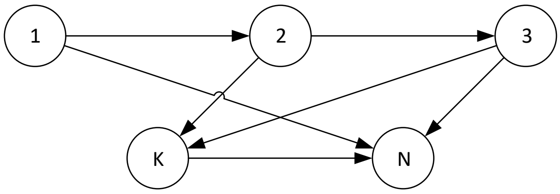

Figure 9: A directed term ontology graph

Usually, the ontology of the terms of a subject area hardly changes – after the ontology is formed, the addition of new concepts and connections between them occurs only with significant changes in the terminology of the subject area.

Thus, it is possible to significantly speed up work on the calculation of semantic distances, if:

1) Ignore terms located far enough away – for example, further than 5-7 transitions from the given term.
2) Create a map of semantic distances to nearby nodes. This method will require additional parameters for the ontology graph – each node must additionally contain data about these semantic distances.

7. Conclusions

This work develops a semantic analysis method based on an inverse-additive metric that describes the semantic distance between ontology terms. The inverse-additive metric allows us to correctly process cases when there are several paths in the directed graph of the ontology from one concept node to another.

The work describes ways to overcome some difficulties in the implementation of this method, associated with critical nodes on the path in the directed ontology graph from one concept node to another. Critical nodes are intersection nodes, merge nodes and branch nodes. The presence of such nodes requires additional analysis of the ontology graph and additional calculations to correctly calculate the semantic distance between ontology concepts.

8. References

[1] Susan T. Dumais, Latent Semantic Analysis, Annual Review of Information Science and Technology 38 (2005) 188–230. doi:10.1002/aris.1440380105.

[2] 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.

[3] 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.

[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] Hryhorovych V, Analysis of Scientific Texts by Semantic Inverse-Additive Metrics for Ontology Concepts, in: Proceedings of the 6th International conference on computational linguistics and intelligent systems (COLINS 2022), Vol. I: main conference. Gliwice, Poland, May 12-13, 2022. – P. 801–815.
[15] Smith, R. J., Circuits, Devices and Systems (International ed.), Wiley, New York, 1966. LCCN 66-17612.
[16] Engineering ToolBox, Electrical Resistance in Serial and Parallel Networks, 2014. URL: https://www.engineeringtoolbox.com/resistance-serial-parallel-connections-d_1881.html .
[17] Wai-Kai Chen, Circuit Analysis and Feedback Amplifier Theory, CRC Press, 2005. ISBN 1420037277.
[18] J. W. Nilsson, S. A. Riedel, Electric Circuits, 8th ed., Pearson Prentice Hall, 2007. ISBN 978-0-13-198925-2.
[19] A. Kumar, N. S. Vidhyadhiraja, G. U. Kulkarni, Current distribution in conducting nanowire networks. Journal of Applied Physics 122.4 (2017). doi:10.1063/1.4985792.
[20] W. H. Hayt, J. E. Kemmerly, (). Engineering Circuit Analysis, 5th ed., McGraw Hill. New York, 1993.
[21] J. W. Nilsson, S. A. Riedel, Introductory Circuits for Electrical and Computer Engineering, Prentice Hall, New Jersey, 2002.