DOI: 10.20998/2079-0023.2024.02.09
UDC 004.94

O. V. RUDSKYI, Student, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: oleksandr.rudskyi@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0001-1130-9957
A. M. KOPP, Doctor of Philosophy (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Head of Software Engineering and Management Intelligent Technologies Department, Kharkiv, Ukraine;
e-mail: andrii.kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623
T. Ye. GONCHARENKO, Candidate of Pedagogical Sciences (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Head of Foreign Languages Department, Kharkiv, Ukraine;
e-mail: tetiana.goncharenko@khpi.edu.ua; ORCID: https://orcid.org/0000-0001-6630-307X
I. P. GAMAYUN, Doctor of Technical Sciences, Professor, National Technical University "Kharkiv Polytechnic Institute", Full Professor of Software Engineering and Management Intelligent Technologies Department, Kharkiv, Ukraine;
e-mail: ihor.hamaiun@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-2099-4658

INTELLIGENT TECHNOLOGY FOR SEMANTIC COMPLETENESS ASSESSMENT OF BUSINESS PROCESS MODELS

In this paper, we present a method for comparing business process models with their textual descriptions, using a semantic-based approach based on the SBERT (Sentence-Bidirectional Encoder Representations from Transformers) model. Business process models, especially those created with the BPMN (Business Process Model and Notation) standard, are crucial for optimizing organizational activities. Ensuring the alignment between these models and their textual descriptions is essential for improving business process accuracy and clarity. Traditional set similarity methods, which rely on tokenization and basic word matching, fail to capture deeper semantic relationships, leading to lower accuracy in comparison. Our approach addresses this issue by leveraging the SBERT model to evaluate the semantic similarity between the text description and the BPMN business process model. The experimental results demonstrate that the SBERT-based method outperforms traditional methods, based on similarity measures, by an average of 31%, offering more reliable and contextually relevant comparisons. The ability of SBERT to capture semantic similarity, including identifying synonyms and contextually relevant terms, provides a significant advantage over simple token-based approaches, which often overlook nuanced language variations. The experimental results demonstrate that the SBERT-based approach, proposed in this study, improves the alignment between textual descriptions and corresponding business process models. This advancement is allowing to improve the overall quality and accuracy of business process documentation, leading to fewer errors, introducing better clarity in business process descriptions, and better communication between all the stakeholders. The overall results obtained in this study contribute to enhancing the quality and consistency of BPMN business process models and related documentation.

Keywords: business process modeling, BPMN, semantic similarity, SBERT, text comparison, business process optimization, natural language processing.

Introduction. In today's world, business process modeling plays an important role in improving management and optimizing organizational activities. However, creating appropriate business process models is a task that requires significant efforts and resources. Comparing business process models with their textual descriptions proves to be a crucial task, as it can help to ensure the accuracy of the model, identify discrepancies, and improve the quality of both the models and the textual descriptions of business processes [1].

In this context, the relevance of evaluating the alignment of business process models with their textual descriptions is evident. Business process modeling, especially using the BPMN standard, provides a tool for representing business processes in a graphical format, making them easier to understand and analyze. However, ensuring consistency between the model and the textual description is essential to avoid errors and inconsistencies in business processes [2].

Comparing business process models with their textual descriptions not only ensures accuracy and consistency but also helps to identify potential shortcomings and ambiguities in the textual descriptions, which can lead to improvements in the quality of business processes. Additionally, this approach fosters a shared understanding among all business process stakeholders, regardless of their level of expertise in process modeling [3].

Related work. A systematic literature review was used to explore current methods for text comparison.

A Systematic Literature Review (SLR) identifies, selects, and critically evaluates studies to answer a clearly formulated question. The systematic review must follow a well-defined protocol or plan that clearly outlines the criteria for conducting the review. It involves a comprehensive and transparent search, conducted across multiple databases and grey literature, which can be replicated by other researchers. This requires a well-thought-out search strategy aimed at answering a specific question. The review identifies the type of information that was searched, critiqued, and reported over a known period of time. Search terms, search strategies (including database names, platforms, search dates), and limitations must all be included in the review [4].

To answer the research questions, the following SLR objectives were defined:

1. Review articles to identify existing methods for text comparison;
2. Highlight weaknesses in the methods with the aim of addressing them through further research;
3. Gain new insights into text comparison methods that can be applied in future research.
4. The following search string was used for the study: ("allintitle:" + "text" + "similarity" + "site:" + "ieeexplore.ieee.org").

Research Article: This article was published by the publishing house of ***NTU "KhPI"*** in the collection "Bulletin of the National Technical University "KhPI" Series: System analysis, management and information technologies." This article is distributed under a Creative Common Creative Common Attribution (CC BY 4.0). **Conflict of Interest:** The author/s declared no conflict of interest.

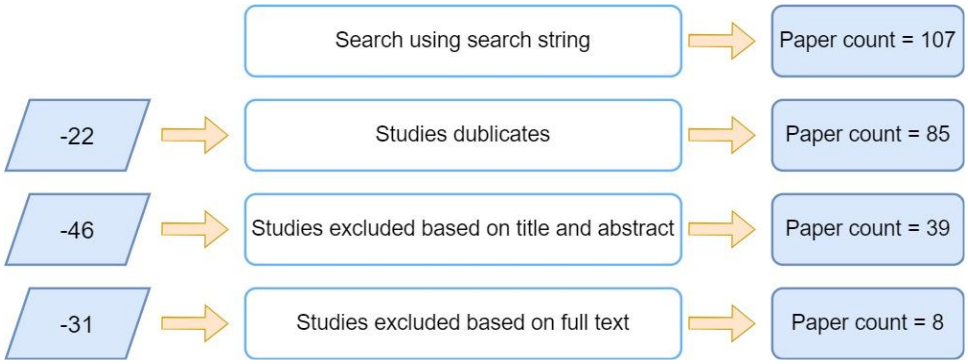

Fig. 1. General systematic literature review scheme

The initial search using only the keywords yielded 107 academic articles related to text similarity in various languages. After thoroughly reviewing the articles, those that were not directly related to the research topic but appeared due to keyword matches were excluded. Additionally, articles were excluded due to duplication, lack of full text, or if the research did not address any of the research questions. After all exclusions, 8 academic articles remained (fig. 1).

State-of-the-art. The first article analyzed was [5], which presents the results of applying various methods for measuring semantic text similarity. The goal of the article is to assess the degree of semantic equivalence of multi-word sentences [5].

One of the methods discussed in [5] is Bag-of-Words (BOW), a technique used to represent fixed-length vectors from which features are extracted for modeling. One of the drawbacks of this method is that the word order is lost, leading to identical vector representations for different sentences with the same words [5].

Another method presented is word2vec. The neural network model "Word2Vec" based on skip-gram predicts surrounding words in sentences without using hidden neurons. Here, the artificial neural network (ANN) is trained on word pairs extracted from documents, considering the window size as a critical parameter of the algorithm. The skip-gram neural network model consists of weights and biases that are updated with each iteration of the input data set, and training on a large set of words would be a time-consuming task [5]. The main idea behind the skip-gram-based Word2Vec algorithm is that a vector is initially randomly initialized for each word in the vocabulary. Then, for each position t, the central word at this position is determined as c, and its context word as o. To identify the context words, a window size of m is defined, meaning that the model will consider words in positions from $t-m$ to $t+m$ as context (fig. 2).

To calculate the probability of a context word by a given central word, each word is represented by two sets of vectors: U_w and V_w. U_w is used when w is a context word, and V_w when w is a central word. Using these two vectors, the probability equation for the central word o and the context word c is as follows:

$$P(O=o \mid C=c)=\frac{\exp\left(u_o^T v_c\right)}{\sum_{w \in Vocab} \exp\left(u_w^T v_c\right)}. \quad (1)$$

In the numerator (1), there is the dot product of words o and c, which reflects the similarity between these two vectors. The higher the similarity, the higher the probability. The denominator (1) normalizes the probability values across the entire vocabulary so that the overall sum equals 1.

The next article analyzed was [6]. This paper discusses a method called Word Vector Distance Decentralization (WVDD), which can handle complex semantic relations, including sentence components and word order [6]. Based on the popular Word2vec model, the WVDD method is proposed for transforming word vectors into sentence vectors and implementing the merging of word vectors to measure sentence similarity, taking into account word order, weighting parameters, and semantic relations. For text clustering, it suggests using the Apache Spark clustering algorithm, which employs the K-means algorithm on the Spark architecture for parallel computing to speed up the text clustering process [6].

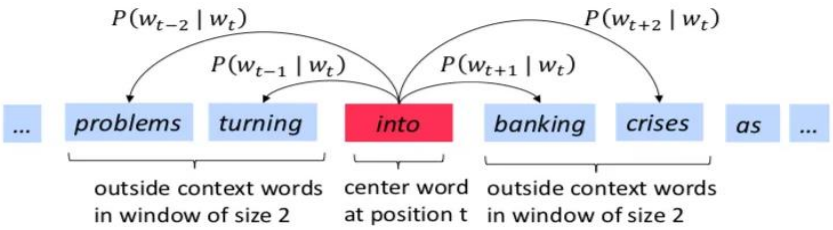

Fig. 2. Process of learning surrounding words in Word2Vec [5]

The following article was [7]. This paper examines the Siamese Neural Network (SNN) and the self2self-attention (S2SA), which is introduced into a Convolutional Neural Network (CNN) to build a new Siamese neural network, specifically the S2SA-SNN.

In S2SA-SNN, self2self-attention is used to learn the varying importance of words and complex syntactic features within a single sentence. Semantic text similarity at the sentence level involves having two sentences. With one sentence X and another sentence Y, the goal of the proposed model is to learn the semantic representations of sentences X and Y and compute a score to measure their similarity or obtain the output of the activation function through these semantic representations.

The next article analyzed was [8]. This paper proposes a short text clustering algorithm based on the fusion of BTM and GloVe similarity (BG & SLF-Kmeans). These are used to model pre-processed short texts. To calculate text similarity based on GloVe word vector modeling, an improved word weighting method (IWMD) is employed. Afterward, the two similarities are linearly combined and used as a distance function to implement clustering via the Kmeans method. The results indicate that BG & SLF-Kmeans significantly improves clustering accuracy compared to TF-IDF & Kmeans, BTM & Kmeans, and BTF & SLF-Kmeans [8].

The next article analyzed was [9]. This paper reviews the limitations of the traditional TF-IDF algorithm and proposes an improved PTF-IDF algorithm. Also, a text classification algorithm based on PTF-IDF and cosine similarity is proposed. Compared to the traditional TF-IDF algorithm, based on an experiment for finding the optimal keyword, the paper finds that text classification accuracy reaches a stable value when the category keywords reach a certain proportion [9].

The next article analyzed was [10]. This paper explores text similarity using a two-stage model for fine-tuning Bidirectional Encoder Representations from Transformers (BERT). Text similarity, as a vertical task in natural language processing, can achieve performance improvements through the two-stage model proposed in this paper [10].

BERT is an abbreviation for Bidirectional Encoder Representation from Transformers, which is a transformer-based machine learning technique for pre-training natural language processing (NLP) developed by Google [10].

BERT can be defined as a function:

$$B: P \to R^{N \times h}, \tag{2}$$

where:

- h is the size of the hidden level;
- $N = 512$ is the maximum sequence length supported by the model.

As an output, BERT (2) receives a paragraph $\rho \in P$ and decomposes it into a sequence $q \in N$ tokens $\left(p^j\right)_{j=1}^{q}$. After that, the sequence (3) is supplemented with N elements by adding special CLS (Classification), SEP (Separator), and PAD (Padded) tokens [10].

This token sequence can be written in the form:

$$I^p = \left(CLS, \left(p^j\right)_{j=1}^{q}, SEP, \ldots, PAD\right). \tag{3}$$

In BERT, all tokens are embedded using three functions: embedded tokens, positions, and segments, denoted as T, O, and G, respectively. Token embedding converts unique token values into intermediate vectors $T\left(I^p\right) \in R^{N \times h}$. Position embedding encodes the token positions into a single space, $O\left(I^p\right) \in R^{N \times h}$. Segment embedding is used to associate each token with one of two sequences $G\left(\{0,1\}^N\right) \in R^{N \times h}$ [10]. The block diagram of the BERT model is given in fig. 3.

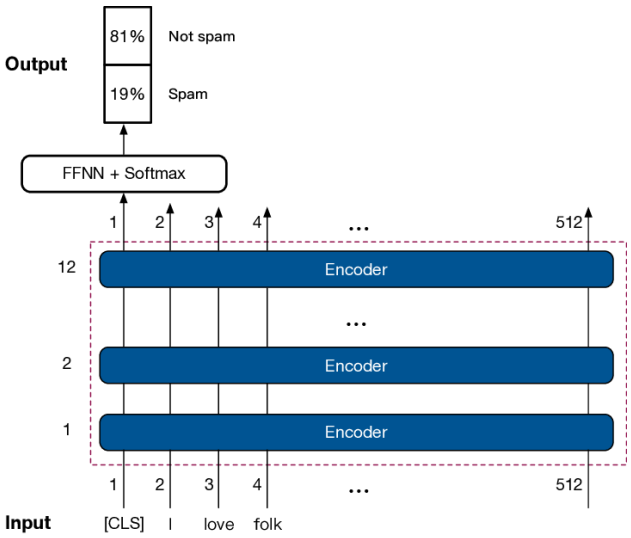

Fig. 3. Block diagram of the BERT model [10]

The next article analyzed was [11]. This paper presents two different models for article categorization. These models consist of two key components: text semantic representation and similarity calculation. First, they represent the text document (article) and then classify it into one of the predefined categories. Afterward, the models dynamically match the output category with the user-defined category. The first model uses TF-IDF features as the semantic representation method, a classifier trained on the BBC dataset, and GloVe to compute category similarity. The second model is an improvement of the first [11]. The GloVe model is an unsupervised learning method used to obtain vector representations of words. GloVe represents words in a multi-dimensional space, placing related words closer to each other in this space. As a result, GloVe implicitly models complex relationships between words in a large vector space. To compute the similarity between different words, GloVe uses cosine similarity and the vector difference between the given words; it associates more than one value for a word pair. The vector difference is needed to better differentiate between words [11].

The next article analyzed was [12]. This paper analyzes the relationship between the true similarity of words and the similarity obtained by various word

embedding methods. The following methods are analyzed in this paper:

1. Word2vec. Word2vec includes two different models: Continuous Bag Of Words (CBOW) and Skip-gram. Both of these methods are neural networks with a hidden layer of N neurons, where N is the dimensionality of the generated word embeddings. The first method, CBOW, is a neural network where the context of words serves as the input. The task is to predict the current word as the network's output. The second method, Skip-gram, is a neural network where the input is a one-hot encoding of a word, and the output is the predicted context of the word, i.e., the surrounding words [12].

2. FastText. The FastText model directly derives from the Skip-gram Word2Vec model. The authors claim that by using a clear vector representation for each word, the Skip-gram model ignores the internal structure of words. For this, they proposed a different scoring function that considers the internal structure. Their subword model represents each word as a bag of character n-grams. Special symbols < and > are added at the beginning and the end of words to distinguish prefixes and suffixes from other character sequences. The word is also included in its set of n-grams to learn a better representation of each word. This model allows sharing representations between words, thus enabling a more robust representation of rare words [12].

3. GloVe (Global Vectors for Word Representation). GloVe is a logarithmic bilinear regression model for unsupervised word representation learning, which combines the advantages of two families of models: global matrix factorization and local context window methods. The overall idea is that the relationship between any two words, i.e., the frequency of words co-occurring in each other's context, encodes information about the words. It captures meaningful linear substructures by effectively using global word co-occurrence statistics. The model is optimized so that the scalar product of any word pair vectors equals the ratio of the corresponding words' occurrences [12].

4. LexVec is based on the idea of factorizing the PPMI matrix using a reconstruction loss function. This loss function does not weigh all errors equally, unlike SVD, but penalizes frequent co-occurrence errors more heavily while also handling negative co-occurrence cases, unlike GloVe. The authors argue that the performance of word similarity and analogy tasks shows that LexVec compares favorably with state-of-the-art methods and often surpasses them in many of these tasks [12].

Algorithm based on the SBERT model. To solve the task of analyzing the alignment of business process models with their textual descriptions, the software application must generate texts T_1 and T_2 based on data extracted from the BPMN and text files. To generate text T_1, the application must extract all the names of “task” elements and related action elements from the BPMN file:

- “Service Task” is a task that uses a service, which can be a web service or an automated application [13];
- “Send Task” is a simple task designed to send a message to an external participant. As soon as the message is sent, the task is completed [13];
- “Receive Task” is a simple task designed to wait for receiving a message from an external user [13];
- “User Task” is a typical task of a business process in which a human executor performs a task with the help of a software application and is scheduled through some task list manager [6];
- “Manual Task” is a task that is supposed to be performed without the help of any business process execution mechanism or any program [13];
- “Business Rule Task” is a task that involves a mechanical process to provide input data for the business rule mechanism and obtain the output data of calculations that the business rule mechanism can provide [13];
- “Script Task” is a task that is executed by the business process engine. When the task is ready to run, the engine will execute the script. After completing the script, the task will also be executed [13].

The following algorithm, presented in the UML activity diagram in fig. 4, can be used to generate text T_1.

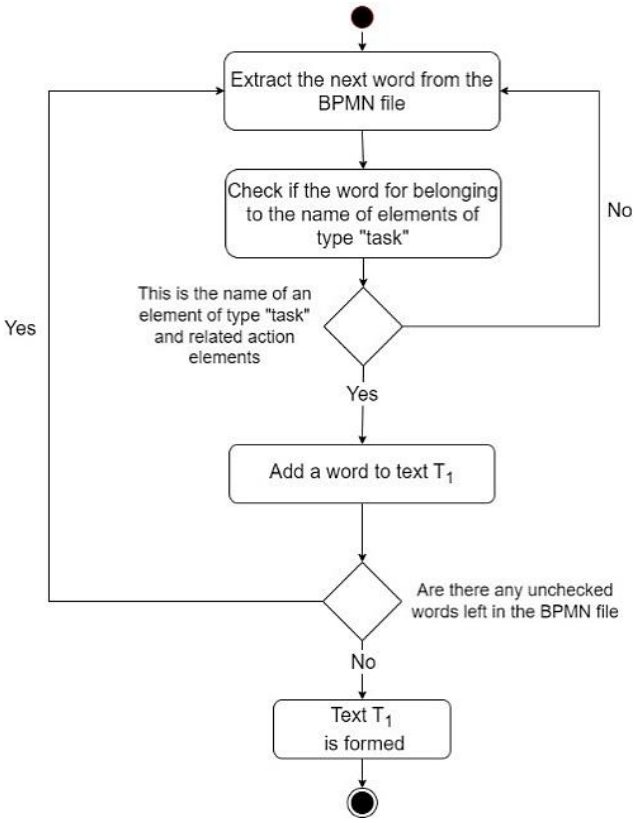

Fig. 4. Algorithm for generating text from the names of business process model tasks

Next, we will use Sentence-BERT (SBERT), a modification of a pre-trained BERT network to measure the degree of semantic textual similarity between two texts.

BERT is used to solve various tasks, such as sentiment analysis or question answering, and it is becoming increasingly popular for creating word embeddings – vector representations of words that reflect their semantic meanings [14].

Representing words as embeddings has provided a huge advantage, as machine learning algorithms cannot work with raw text but can work with vector embeddings.

This allows different words to be compared based on their similarity using standard metrics, such as Euclidean or cosine similarity [14].

Transformer-based models expect a sequence of tokens as input. Therefore, the very first step is to transform the input text into a sequence of tokens, or tokenization. BERT accepts the token [CLS] and two sentences separated by a special [SEP] token as input. Depending on the maximum token sequence length, which is predetermined, a set of [PAD] tokens will also be added after the [SEP] token. Depending on the model configuration, this information is processed 12 or 24 times by multi-head attention blocks. The output is then aggregated and passed to a simple regression model to produce the final label [15]. fig. 5 shows the architecture of the BERT model.

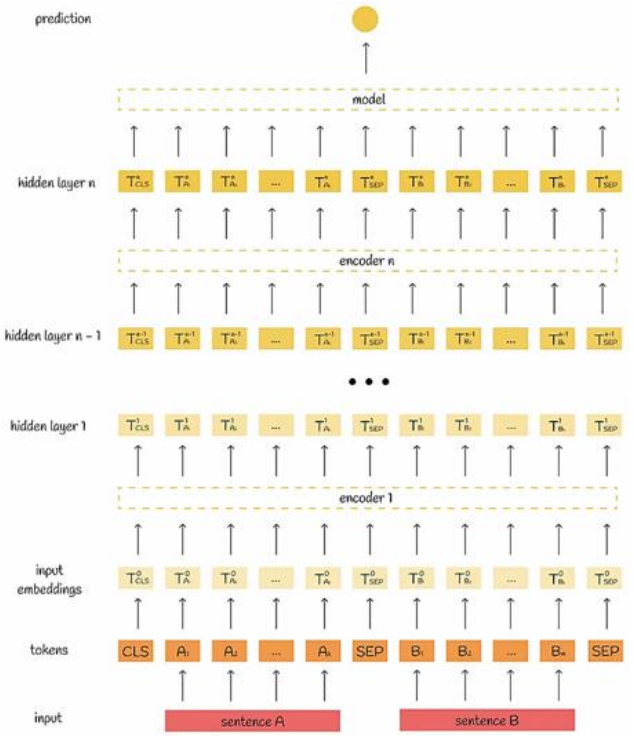

Fig. 5. Architecture of the BERT model [15]

The main problem with BERT is that whenever two sentences are passed and processed simultaneously, it complicates obtaining embeddings that independently represent only one sentence [14].

SBERT introduces the concept of a Siamese network, which means that two sentences are independently pass through the same BERT model each time. The Siamese network architecture allows splitting fixed-size vectors for the input sentences [14].

Fig. 6 shows a comparison between the non-Siamese and Siamese architectures. As can be seen in the figure, the key difference is that on the left, the model processes both inputs simultaneously, while on the right, the model processes both inputs in parallel, meaning the outputs are independent of each other.

After the sentence passes through BERT, a pooling layer is applied to the BERT embeddings to obtain a lower-dimensional representation: the initial 512 768-dimensional vectors are converted into a single 768-dimensional vector. Mean pooling is chosen for the pooling layer [14].

Once both sentences are passed through the pooling layers, we obtain two 768-dimensional vectors, u and v (fig. 7). After obtaining the vectors u and v, the similarity between them is directly computed using cosine similarity. The predicted similarity score is compared with the true value, and the model is updated using the MSE loss function [14]. Fig. 7 presents the SBERT architecture for calculating the similarity score.

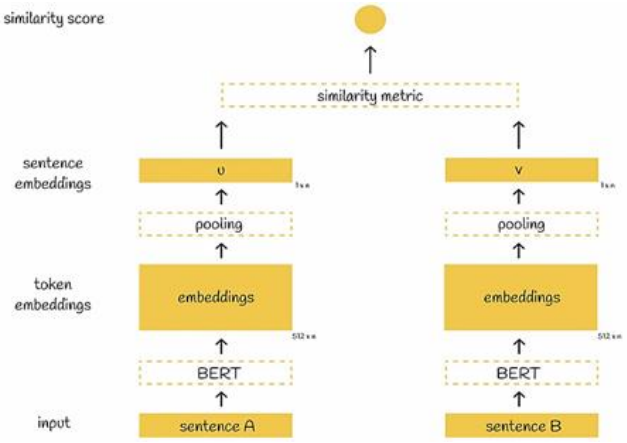

Fig. 7. SBERT architecture for similarity score calculation [15]

By using a similarity measure such as cosine similarity, semantically similar sentences can be found:

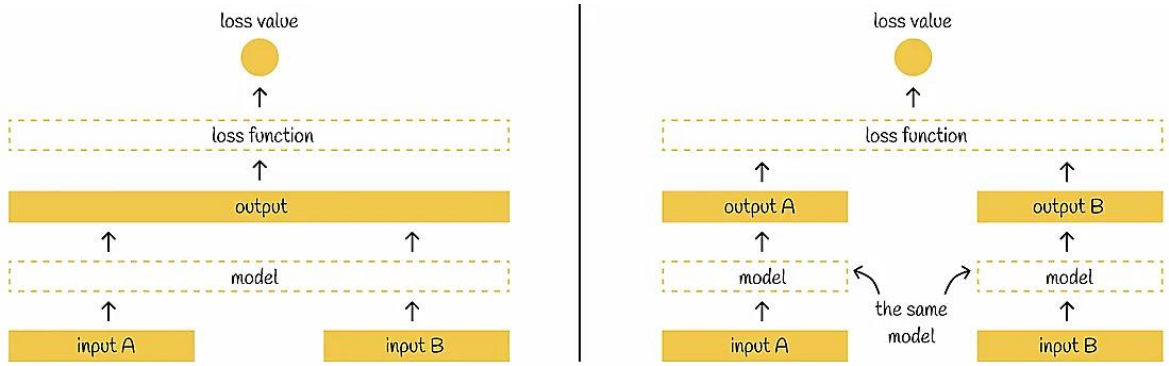

Fig. 6. Comparison of non-Siamese and Siamese architectures [15]

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (4)$$

where A_i and B_i are coordinates of vectors A and B, respectively.

This similarity measure (4) can be efficiently computed on modern hardware, allowing SBERT to be used for both semantic similarity search and clustering [14].

Results and discussion. To evaluate the effectiveness of the proposed method, it was compared to the set similarity method, which consists of the following steps:

- tokenization;
- stop-word removal;
- word stemming.

The similarity of word sets in this method is calculated using the Jaccard coefficient.

The performance of these methods was tested on four business process models.

The first model considered is the business process called "Dispatch of goods" [16]. The text of this process is provided below, and the BPMN model of the business process is shown in fig. 8. First text: "If goods shall be shipped, the secretary clarifies who will do the shipping. If you have large amounts, special shipping will be necessary. In these cases, the secretary invites three logistic companies to make offers and she selects one of them. In case of small amounts, normal post shipment is used. Therefore, a package label is written by the secretary and a parcel insurance taken by the logistics department head if necessary. In the meantime, the goods can be already packaged by the warehousemen. If everything is ready, the packaged goods are prepared for being picked up by the logistic company".

Based on this model, the following names of the tasks were defined:

- "Insure parcel";
- "Write package label";
- "Clarify shipment method";
- "Get 3 offers from logistic companies";
- "Select logistic company and place order";
- "Package goods";
- "Prepare for picking up goods".

The first method showed a result of 38% similarity.

The proposed new method showed a result of 72% similarity.

The next model considered is "Credit Scoring Asynchronous" [16]. The text of this process is provided below, and the BPMN model of the business process is shown in fig. 9. Second text: "The sales clerks in a bank can use their software frontend to receive the credit-scoring for a certain customer. This starts a process in the banking system which communicates with the agency in the background. This process sends a scoring request to the agency right after the beginning. Then, the Agency does a first quick scoring (level 1). This will often lead to an immediate result which is then returned directly to the banking system within seconds. The banking process presents the result to the clerk sitting at the frontend. Sometimes the scoring cannot be determined immediately and takes longer. In this case the agency informs the banking process of the delay and then starts the level 2 scoring (which can take up to a couple of minutes). After the scoring result is determined, the information is sent back to the banking process. The banking process displays a message to the clerk when he receives information about the delay to check again later. As soon as the result arrives, it can be seen at the frontend".

Based on this model, the following names of the tasks were defined:

- "Request credit score";

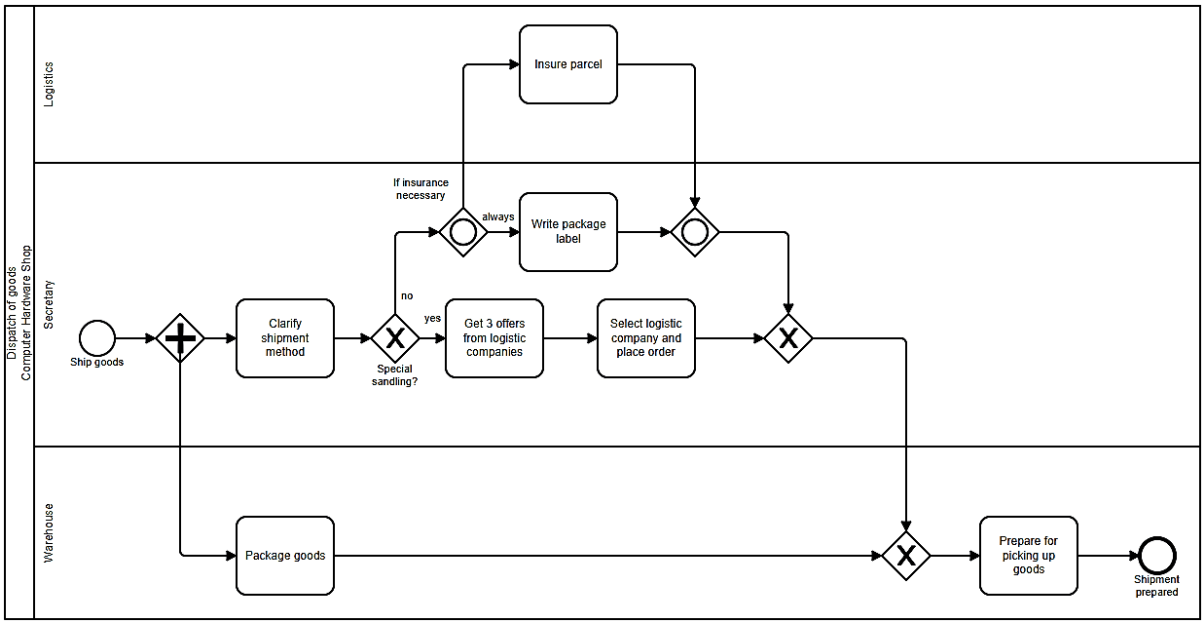

Fig. 8. Model 1 – "Dispatch of goods"

- “Send credit score”;
- “Report delay”;
- “Send credit score”;
- “Report delay”;
- “Send credit score”;
- “Compute credit score (level 2)”;
- “Compute credit score (level 1)”.

The first method showed a result of 12% similarity.

The proposed new method showed a result of 58% similarity.

The next model considered is “Recourse” [16]. The text of this process is provided below, and the BPMN model of the business process is shown in fig. 10. Third text: “If an insurant could be possibly subrogated against, I get information about that. I check that case and if the possibility is really there, I send a request for payment to the insurant and make me a reminder. If recourse is not possible, I close the case. When we receive the money, I make a booking and close the case. If the insurant disagrees with the recourse, I will have to check the reasoning of that. If he is right, I simply close the case. If he is wrong, I forward the case to a collection agency. It the deadline for disagreement is reached and we have not received any money, I forward the case to the collection agency as well”.

Based on this model, the following names of the tasks were defined:

- “Check case”;
- “Send request for payment”;
- “Close case”;
- “Send reminder”;
- “Check reasoning”;
- “Close case”;
- “Hand over to collection agency”;
- “Make booking”;
- “Close case”.

The first method showed a result of 44% similarity.

The proposed new method showed a result of 61% similarity.

The next model considered is “Self Service Restaurant” [16]. The text of this process is provided

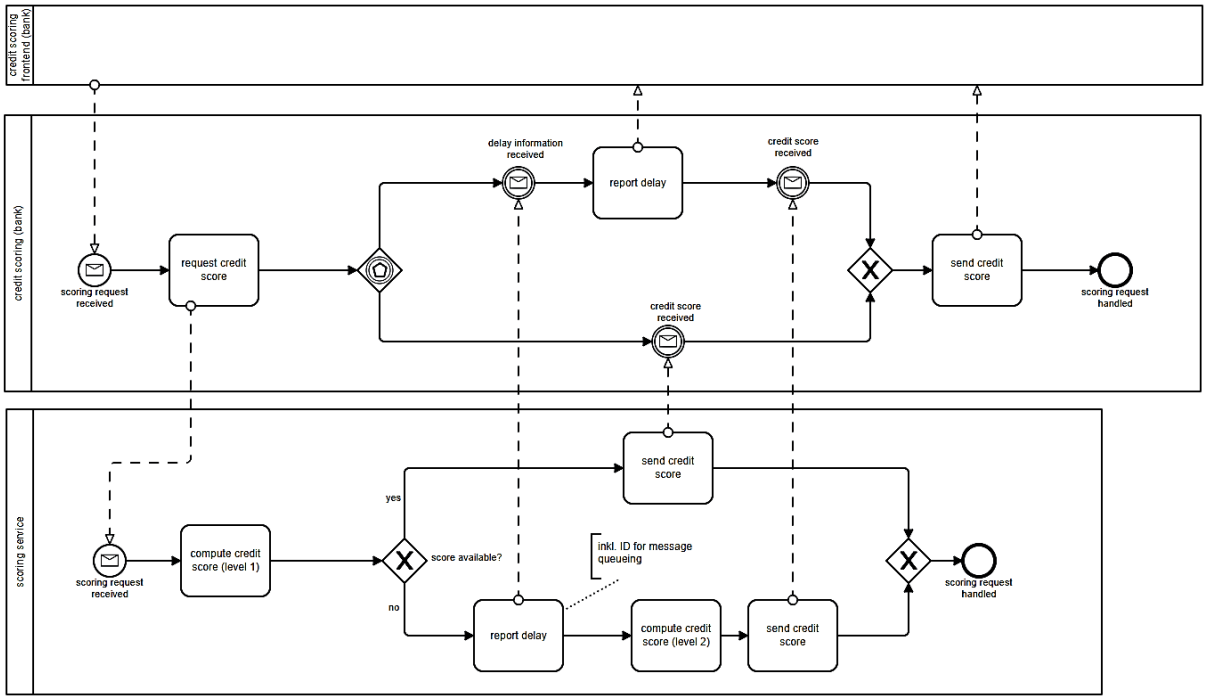

Fig. 9. Model 2 – “Credit Scoring Asynchronous”

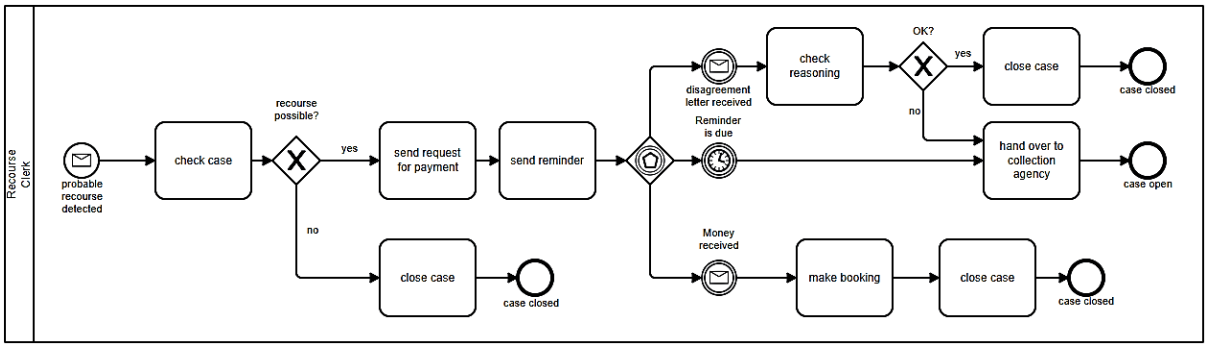

Fig. 10. Model 3 – “Recourse”

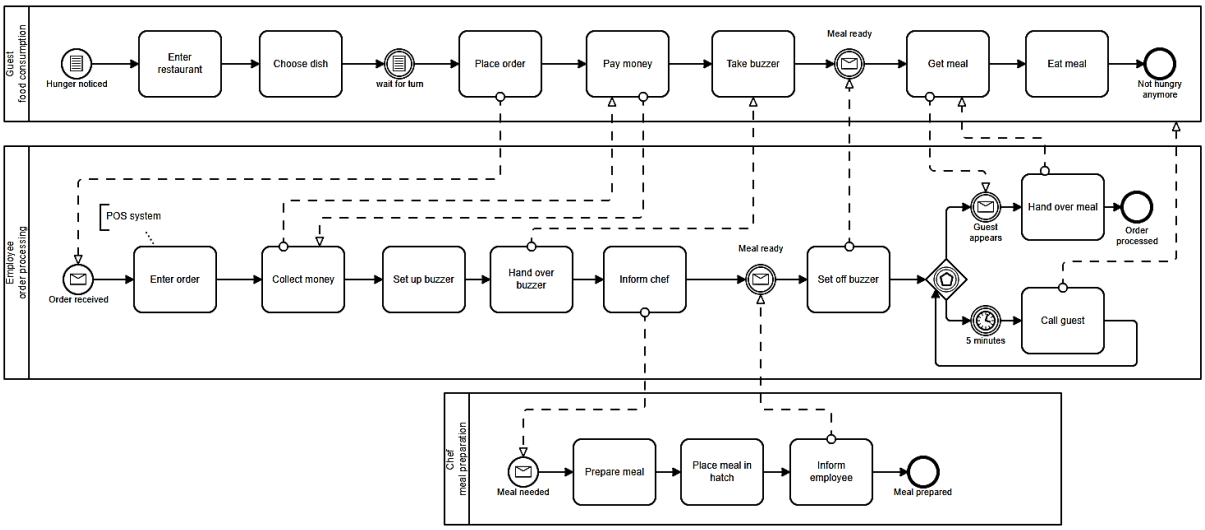

Fig. 11. Model 4 – "Self Service Restaurant"

below, and the BPMN model of the business process is shown in fig. 11. Fourth text: "A guest enters the restaurant when feeling hungry. He chooses a dish from the changing meal range and waits until it is his turn. Following this he places his order with the employee. The employee enters the order into the POS system and collects the money from the guest. After the payment, the employee sets up a buzzer and passes it on to the guest with the following information: When the buzzer rings, your dinner is ready. Afterwards the employee informs the chef of the new meal order. The chef prepares the meal and places it in the service hatch. He then informs the employee that he has placed the finished meal in the service hatch. As soon as the employee is aware that the meal is ready he sets off the guests buzzer. This is how the guest finds out that his meal is ready for collection. He can pick up his meal and eat it. As soon as the guest appears at the service hatch, the employee hands over his meal. Should a guest not react to the buzzer, the employee calls for him after 5 minutes, if necessary several times in a row".

Based on this model, the following names of the tasks were defined:

- "Enter restaurant";
- "Choose dish";
- "Place order";
- "Pay money";
- "Take buzzer";
- "Get meal";
- "Eat meal";
- "Enter order";
- "Collect money";
- "Set up buzzer";
- "Hand over buzzer";
- "Inform chef";
- "Set off buzzer";
- "Hand over meal";

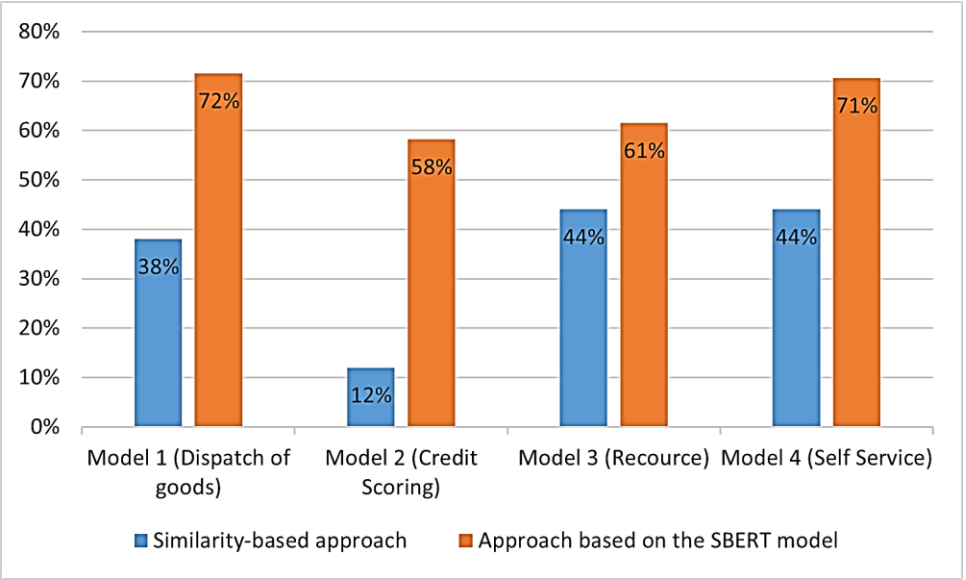

Fig. 12. Comparison results chart

- "Call guest";
- "Prepare meal";
- "Place meal in hatch";
- "Inform employee".

The first method showed a result of 44% similarity.

The proposed new method showed a result of 71% similarity.

Fig. 12 shows a bar chart with the comparison results.

As can be seen in fig. 12, the SBERT-based approach provides an average of 31% higher similarity compared to the set similarity approach. This is due to the fact that the set similarity approach cannot identify synonym words and semantic similarity, leading to a lower comparison score.

Conclusion and future work. In this paper, we have demonstrated the effectiveness of using a semantic-based approach for comparing business process models and their textual descriptions. The proposed method, based on SBERT, outperforms the traditional set similarity approach by an average of 31%, as shown in the comparative analysis of multiple business process models. The ability of SBERT to capture semantic similarity, including identifying synonyms and contextually relevant terms, provides a significant advantage over simple token-based approaches, which often overlook nuanced language variations.

The experimental results show that the SBERT-based approach improves the alignment of textual descriptions with business process models. This advancement enhances the overall quality and accuracy of business process documentation, leading to fewer errors, more clarity in process descriptions, and better communication between stakeholders.

In the future, we plan to improve our work with industry-specific terminology, which will allow for more accurate comparisons of models in specialized sectors. In addition, we plan to explore real-time applications of this method, such as using semantic analysis tools during the modeling process to provide immediate feedback on discrepancies between the BPMN model and its textual description.

References

1. Jošt G., Polančič G., Heričko M., Kocbek M. *Business process model and notation: The current state of affairs*. URL: https://doi.org/10.2298/CSIS140610006K (access date: 20.09.2024).
2. Von Rosing M., White S., Cummins F., De Man H. *Business process model and notation-BPMN*. URL: https://doi.org/10.1016/B978-0-12-799959-3.00021-5 (access date: 20.09.2024).
3. Mroczek A., Wiśniewski P., Ligęza A. *Overview of Verification Tools for Business Process Models*. URL: https://doi.org/10.15439/2017f308 (access date: 20.09.2024).
4. Ottensooser A., Fekete A., Reijers H., Mendling J., Menictas C. *Making sense of business process descriptions: An experimental comparison of graphical and textual notations*. URL: https://doi.org/10.1016/j.jss.2011.09.023 (access date: 20.09.2024).
5. Qurashi A., Holmes V., Johnson A. *Document Processing: Methods for Semantic Text Similarity Analysis*. URL: https://doi.org/10.1109/INISTA49547.2020.9194665 (access date: 20.09.2024).
6. Zhou S., Xu X., Liu Y., Chang R., Xiao Y. *Text Similarity Measurement of Semantic Cognition Based on Word Vector Distance Decentralization with Clustering Analysis*. URL: https://doi.org/10.1109/ACCESS.2019.2932334 (access date: 20.09.2024).
7. Li Z., Chen H., Chen H. *Biomedical Text Similarity Evaluation Using Attention Mechanism and Siamese Neural Network*. URL: https://doi.org/10.1109/ACCESS.2021.3099021 (access date: 20.09.2024).
8. Wu D., Zhang M., Shen C., Huang Z., Gu M. *BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery*. URL: https://doi.org/10.1109/ACCESS.2020.2973430 (access date: 20.09.2024).
9. Liu Y., Xu Q., Tang Z. *Research on Text Classification Method Based on PTF-IDF and Cosine Similarity*. URL: https://doi.org/10.1109/ICIIBMS46890.2019.8991542 (access date: 20.09.2024).
10. Zhengfang H., MacHica I., Zhimin B. *Textual Similarity Based on Double Siamese Text Convolutional Neural Networks and Using BERT for Pre-training Model*. URL: https://doi.org/10.1109/ICAIBD55127.2022.9820371 (access date: 20.09.2024).
11. Dauzhan G., Iskakov A., Kenzhegaliyev M., Bui D. *Dynamic Text Modeling and Categorization Framework based on Semantics Extraction and Similarity Checking*. URL: https://doi.org/10.1109/CSCI58124.2022.00132 (access date: 20.09.2024).
12. Toshevska M., Stojanovska F., Kalajdjieski J. *Comparative Analysis of Word Embeddings for Capturing Word Similarities*. URL: https://doi.org/10.5121/csit.2020.100402 (access date: 20.09.2024).
13. *Business Process Model and Notation (BPMN), Version 2.0*. URL: https://www.omg.org/spec/BPMN/2.0/PDF (access date: 20.09.2024).
14. Reimers N., Gurevych I. *Sentence-BERT: Sentence embeddings using siamese BERT-Networks*. URL: https://doi.org/10.18653/v1/d19-1410 (access date: 20.09.2024).
15. *Large Language Models: SBERT – Sentence-BERT*. URL: https://towardsdatascience.com/sbert-deb3d4aef8a4 (access date: 20.09.2024).
16. *BPMN for research*. URL: https://github.com/camunda/bpmn-for-research (access date: 20.09.2024).

References (transliterated)

1. Jošt G., Polančič G., Heričko M., Kocbek M. *Business process model and notation: The current state of affairs*. Available at: https://doi.org/10.2298/CSIS140610006K (accessed: 20.09.2024).
2. Von Rosing M., White S., Cummins F., De Man H. *Business process model and notation-BPMN*. Available at: https://doi.org/10.1016/B978-0-12-799959-3.00021-5 (accessed: 20.09.2024).
3. Mroczek A., Wiśniewski P., Ligęza A. *Overview of Verification Tools for Business Process Models*. Available at: https://doi.org/10.15439/2017f308 (accessed: 20.09.2024).
4. Ottensooser A., Fekete A., Reijers H., Mendling J., Menictas C. *Making sense of business process descriptions: An experimental comparison of graphical and textual notations*. Available at: https://doi.org/10.1016/j.jss.2011.09.023 (accessed: 20.09.2024).
5. Qurashi A., Holmes V., Johnson A. *Document Processing: Methods for Semantic Text Similarity Analysis*. Available at: https://doi.org/10.1109/INISTA49547.2020.9194665 (accessed: 20.09.2024).
6. Zhou S., Xu X., Liu Y., Chang R., Xiao Y. *Text Similarity Measurement of Semantic Cognition Based on Word Vector Distance Decentralization with Clustering Analysis*. Available at: https://doi.org/10.1109/ACCESS.2019.2932334 (accessed: 20.09.2024).
7. Li Z., Chen H., Chen H. *Biomedical Text Similarity Evaluation Using Attention Mechanism and Siamese Neural Network*. Available at: https://doi.org/10.1109/ACCESS.2021.3099021 (accessed: 20.09.2024).
8. Wu D., Zhang M., Shen C., Huang Z., Gu M. *BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery*. Available at: https://doi.org/10.1109/ACCESS.2020.2973430 (accessed: 20.09.2024).
9. Liu Y., Xu Q., Tang Z. *Research on Text Classification Method Based on PTF-IDF and Cosine Similarity*. Available at: https://doi.org/10.1109/ICIIBMS46890.2019.8991542 (accessed: 20.09.2024).
10. Zhengfang H., MacHica I., Zhimin B. *Textual Similarity Based on Double Siamese Text Convolutional Neural Networks and Using*

BERT for Pre-training Model. Available at: https://doi.org/10.1109/ICAIBD55127.2022.9820371 (accessed: 20.09.2024).
11. Dauzhan G., Iskakov A., Kenzhegaliyev M., Bui D. *Dynamic Text Modeling and Categorization Framework based on Semantics Extraction and Similarity Checking*. Available at: https://doi.org/10.1109/CSCI58124.2022.00132 (accessed: 20.09.2024).
12. Toshevska M., Stojanovska F., Kalajdjieski J. *Comparative Analysis of Word Embeddings for Capturing Word Similarities*. Available at: https://doi.org/10.5121/csit.2020.100402 (accessed: 20.09.2024).
13. *Business Process Model and Notation (BPMN), Version 2.0*. Available at: https://www.omg.org/spec/BPMN/2.0/PDF (accessed: 20.09.2024).
14. Reimers N., Gurevych I. *Sentence-BERT: Sentence embeddings using siamese BERT-Networks*. Available at: https://doi.org/10.18653/v1/d19-1410 (accessed: 20.09.2024).
15. *Large Language Models: SBERT – Sentence-BERT*. Available at: https://towardsdatascience.com/sbert-deb3d4aef8a4 (accessed: 20.09.2024).
16. *BPMN for research*. Available at: https://github.com/camunda/bpmn-for-research (accessed: 20.09.2024).

Received 15.11.2024

УДК 004.94

О. В. РУДСЬКИЙ, Національний технічний університет «Харківський політехнічний інститут», студент, м. Харків, Україна; e-mail: oleksandr.rudskyi@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0001-1130-9957
А. М. КОПП, доктор філософії (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: andrii.kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623
Т. Є. ГОНЧАРЕНКО, кандидат педагогічних наук (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри іноземних мов, м. Харків, Україна; e-mail: tetiana.goncharenko@khpi.edu.ua; ORCID: https://orcid.org/0000-0001-6630-307X
І. П. ГАМАЮН, доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», професор кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: ihor.hamaiun@khpi.edu.ua; ORCID: https://orcid.org/0000-0003-2099-4658

ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ ОЦІНЮВАННЯ СЕМАНТИЧНОЇ ПОВНОТИ МОДЕЛЕЙ БІЗНЕС-ПРОЦЕСІВ

У цій статті авторами представлено метод порівняння моделей бізнес-процесів з їх текстовими описами на основі використання семантичного підходу з використанням моделі SBERT (Sentence-Bidirectional Encoder Representations from Transformers). Моделі бізнес-процесів, зокрема створені за стандартом BPMN (Business Process Model and Notation), мають вирішальне значення для оптимізації організаційної діяльності. Забезпечення узгодженості між цими моделями та їхніми текстовими описами має важливе значення для підвищення точності та зрозумілості бізнес-процесів. Традиційні методи схожості множин, які покладаються на токенізацію та базове зіставлення слів, не можуть охопити глибші семантичні зв'язки, що призводить до нижчої точності порівняння. Запропонований підхід дозволяє розв'язати цю задачу, за рахунок використання моделі SBERT для оцінки семантичної подібності між текстовим описом і BPMN-моделлю бізнес-процесу. Експериментальні результати демонструють, що метод на основі SBERT перевершує традиційні методи, засновані на показниках подібності, в середньому на 31%, пропонуючи більш надійні та контекстуально відповідні порівняння. Здатність SBERT фіксувати семантичну схожість, включаючи ідентифікацію синонімів і контекстуально релевантних термінів, забезпечує значну перевагу перед більш простими підходами на основі токенізації, які часто не помічають нюансів мовних варіацій. Експериментальні результати демонструють, що підхід на основі SBERT, запропонований у цьому дослідженні, покращує узгодженість між текстовими описами та відповідними моделями бізнес-процесів. Таке удосконалення дозволяє підвищити загальну якість і точність документації бізнес-процесів, що призводить до зменшення помилок, запровадження кращої зрозумілості описів бізнес-процесів, а також кращої взаємодії між усіма зацікавленими сторонами. Загальні результати, отримані в цьому дослідженні, сприяють підвищенню якості та узгодженості моделей бізнес-процесів BPMN і відповідної документації.

Ключові слова: моделювання бізнес-процесів, BPMN, семантична подібність, SBERT, порівняння текстів, оптимізація бізнес-процесів, обробка природної мови.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Рудський Олександр Вадимович / Rudskyi Oleksandr Vadymovych
Автор 2 / Author 2: Копп Андрій Михайлович / Kopp Andrii Mykhailovych
Автор 3 / Author 3: Гончаренко Тетяна Євгенівна / Goncharenko Tetiana Yevhenivna
Автор 4 / Author 4: Гамаюн Ігор Петрович / Gamayun Igor Petrovych