Intelligent System for Semantically Similar Sentences Identification and Generation Based on Machine Learning Methods

Petro Zdebskyi[0000-0002-0478-2308]1, Vasyl Lytvyn[0000-0002-9676-0180]2, Yevhen Burov[0000-0001-8653-1520]3, Zoriana Rybchak[0000-0002-5986-4618]4, Petro Kravets[0000-0001-8569-423X]5, Olga Lozynska[0000-0003-2882-3546]6, Roman Holoshchuk[0000-0002-1811-3025]7, Solomiya Kubinska[0000-0003-3201-635X]8, Alina Dmytriv[0000-0003-0141-6617]9

Lviv Polytechnic National University, Lviv, Ukraine

petrozd@gmail.com1, Vasyl.V.Lytvyn@lpnu.ua2, Yevhen.V.Burov@lpnu.ua3, zozylka3@gmail.com4, Petro.O.Kravets@lpnu.ua5, Olha.V.Lozynska@lpnu.ua6, roman@ridne.net7, kubinskasm@gmail.com8, alinadmutriv@gmail.com9

Abstract. The task of generating semantically similar sentences can be reduced to the task of generating text and verifying that the generated text is semantically similar to the sample. This article describes all the main technical aspects of solving this problem, describes proposed solutions for the development of algorithmic, functional and software components of the application of identification and generation of semantically similar sentences. During the analysis of existing algorithms, the basic principles of operation of such algorithms were considered. Analogues were analyzed, namely the methods of semantic comparison of sentences, their advantages and disadvantages were determined. The methods that solve the problem are many, but they have some limitations, such as unreliability after slight changes to the text or paraphrase. This article describes the software implementation of the task. Different ways of semantic comparison and text generation are analyzed. Also, the system was tested for new data, that is, data that was not used to train the model.

Keywords. Machine learning, intelligent system, semantically similar sentences

1 Introduction

Formally, the task of identifying semantically similar sentences can be considered as a task of "Recognizing Textual Entailment". The Recognizing Textual Entailment (RTE) is a task of recognizing two pieces of text, or the value of one can be deduced from the other. This task is not domain-specific, and it is proposed to recognize the variability of semantic expression that is commonly required in many tasks [1].

The fundamental phenomenon of natural language is the variety of semantic expressions in which the same meaning can be expressed or logically derived from dif-

ferent texts. This phenomenon can be seen as a problem of linguistic ambiguity, which links many to many between linguistic expressions and meanings [1-2].

The textual relation of logical inference between two texts: T (text) and H (hypothesis) represents a fundamental phenomenon of natural language. This is denoted as T → H and means that the value of H can be logically deduced from T [3].

This relation is directed because the value of one expression (e.g. "buy") can usually be logically deduced from another (e.g. "own") and the logical inference to the other side is less obvious [2]. Text-based logical inference is context-sensitive, non-transitive, and non-monotonous. [4]

2 Literature analysis

2.1 Application of Identification of Semantically Similar Sentences

Text Logic Recognition is one of the most complex natural language processing tasks, and progress in this task is key to solving other tasks, such as Question Answering, Information Extraction, Information Retrieval, Text Summarization, and more. For example, a system of answering a question should identify the text that is logically displayed as the expected answer. Given the question, the text is logically derived from the expected answer. Similarly, in the search for information, the request should be logically derived from the documents received. In the summarization, excess sentences can be omitted if they can be logically deduced from other sentences. In the task of retrieving information, logical output is between different variants of text that express the same relation to the target text. In a machine translation check, the correct translation must be semantically similar to the model translation and should therefore be logically deducible from each other. Therefore, in the same way in the Word Sense Disambiguation task, which is regarded as a common task, solving logical inference can consolidate research in the application of semantic inference [3].

2.2 Application of Generating Semantically Similar Sentences

The mechanism of automatically generating different paraphrases of a single sentence will have a great practical impact on text generation systems that accept text as input and output text. Applied tasks include summarizing and rewriting text. Another interesting application is the use of generating semantically similar sentences to expand datasets by adding multiple versions of their sentences. This is useful for both machine translation and so-called data augmentations, which are used to train machine learning models.

2.3 Description of the Subject Area in Terms of Ontological Engineering

The recent rapid progress of neural network natural language research, especially in the study of semantic textual images, can allow for truly new products. It can also

help improve performance on a variety of natural language tasks that have a limited amount of training data, such as building strong text classifiers from just 100 examples. Building ontology of a particular domain is now based on the intuition of a knowledge engineer, and the typical output is a thesaurus of terms, each of which is expected to denote the concept. Ontology engineers typically design a thesaurus on a special basis and on a relatively small scale. Workers in a particular domain create their own custom language, and one device for that creation is to repeat the selected keywords to consolidate, or to reject, one or more concepts. A more scalable, systematic, and automated approach to ontology construction is made possible by the automatic identification of these keywords. Keyword learning and retrieval approaches are used to analyze the corpus of randomly collected unstructured, that is, does not contain any type of markup, texts in a specific area, with reference to lexical preferences of employees in the domain. Analyzing commonly used words in word combinations leads to the creation of a semantic network. The network can be introduced into a terminological database or formalism of knowledge representation and the interconnection between the nodes of the network helps in the visualization and automatic output of commonly used words denoting important concepts in the field [1].

2.4 Learning Semantic Textual Similarity from Conversations

An approach to the semantic similarity of the sentence level is based on the uncontrolled study of spoken data. The intuition is that sentences are semantically similar if they have a similar distribution of responses, and that the model trained to predict the relationships between inputs must implicitly learn useful semantic representation. For example, “How old are you?” and “What is your age?” are both questions about age, which can be answered by similar responses such as “I am 20 years old”. In contrast, while “How are you?” and “How old are you?” contain almost identical words, they have very different meanings and lead to different responses. (Fig. 1) [2].

Fig. 1. Sentences are semantically similar if they can be answered by the same responses. Otherwise, they are semantically different [2]

2.5 Universal Sentence Encoder

Methods for generating vector representation of words include neural networks [3] dimensionality reduction on adjacent word prevalence matrices, [4-6] probabilistic

models [7] explanatory knowledge base method [8], and explicit representation in terms of context in which words appear [9-15].

The representations of words computed using neural networks are very interesting because the vectors obtained encode many language patterns and patterns. Somewhat surprisingly, many of these structures can be represented as linear transformations.

Many natural language learning tasks have limited amounts of training data available [16-19]. This is a challenge for deep learning methods. Given the high cost of annotated, supervised data, very large training kits are usually not available for most research and industry tasks. Universal Sentence Encoder is a model that extends multitasking learning by adding more tasks, the model tries to surround the text by getting some text input. However, instead of the encoder-decoder architecture in the original model, the encoder-only architecture was used by coding jointly to manage prediction tasks [21-27]. So, learning time is greatly reduced, maintaining performance on a variety of tasks, including mood classification and semantic similarity (Fig 2).

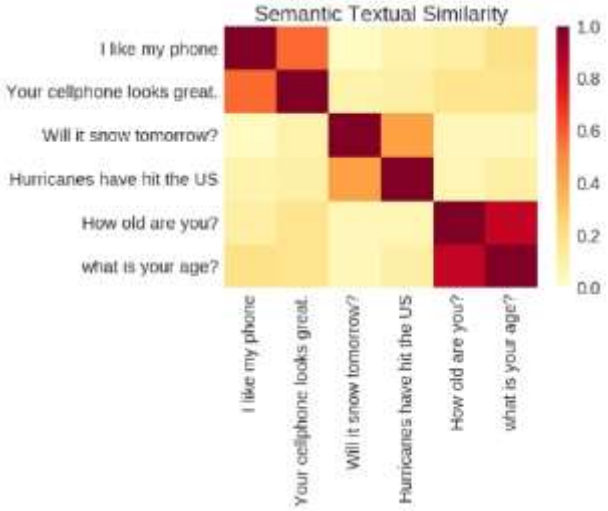

Fig. 2. Comparison of semantic similarity between pairs of sentences with Universal Phrase Encoder [2]

The aim is to provide a single encoder that can support as wide a variety of applications as possible, including paraphrase detection, relatedness, clustering and custom text classification (Fig 3) [28-32]. The two versions of Universal Sentence Encoder use different architectures. The simpler version, which runs faster but uses Deep Average Network (DAN) with slightly less accuracy, the more complex version uses the Transformer architecture.

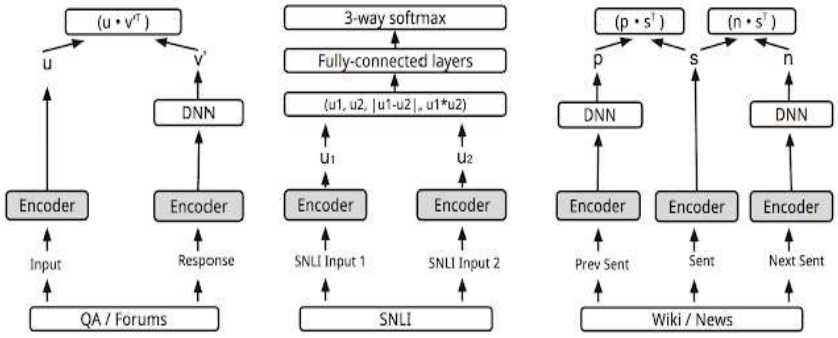

Fig. 3. Using Universal Sentence Encoder for a variety of tasks

2.6 Scheme of Interconnections

According to the phrase structure, the grammar of the sentence consists of a noun phrase and a verb phrase [33-37]. The scheme is presented in Fig. 4.

Fig. 4. Scheme of interconnections according to the phrase structure of the sentence in English.

2.7 Natural Language Processing

Natural Language Processing (NLP) is a general field of computer science, artificial intelligence and mathematical linguistics. He studies the problems of computer analysis and natural language synthesis. For artificial intelligence, analysis means understanding the language and synthesis means generating intelligent text. Solving these problems will mean creating a more convenient form of computer-human interaction.

According to researcher Elizabeth Liddy: "Natural language processing is a computerized approach to text analysis based on a number of theories and a set of technologies. This industry does not have one accepted definition, as it is in a state of constant research and development. However, there are certain aspects that would integrate all existing definitions [38-43].

Tasks and Limitations

Natural language understanding is sometimes considered an AI-complete task, because recognizing a living language requires vast knowledge of the environment and the ability to interact with it. The very definition of the meaning of the word "understand" is one of the main tasks of artificial intelligence. Nowadays, ontologies, such as WordNet, UWN, play a significant role in the processing of natural-language data processing tasks. Significant results have been achieved in the study of natural lan-

guage processing, including the development of powerful lexicographic systems, machine translation programs, electronic dictionaries and more. However, there is a problem that has not yet been resolved, it is rooted in the very nature of human language. The problem of understanding human speech lies precisely in its ambiguity. The following types of ambiguities can be distinguished [44-51]:

1. Syntactic ambiguity: in the adverb "Time is not a horse, you will not stop and you will not stop" for the processing of natural language it will be absolutely unclear what is said in the sentence, about the horse or about time.
2. The semantic ambiguity: in the question "Where to find the key to that castle?" The word castle can have two completely different meanings, given the emphasis.
3. Great ambiguity: in the words "Everyone was excited before the concert" and "Don't give it before!" The word before means a time or place that completely changes the meaning of the phrase.
4. Reference ambiguity: in the phrase "Open the shelf and get the wet umbrella, I want to dry it" her pronoun meaningfully refers to the wet umbrella, but for a machine that has no complete understanding of reality, this pronoun applies to both the shelf and the shelf umbrellas.

One of the challenges that arise in the process of natural language processing can be considered to be the problem of synonymy, whereby one concept can be expressed in several different words. As a consequence, relevant documents that use synonyms of the terms specified by the user in the request may not be identified by the system.

The impact of the above phenomena is particularly noticeable when creating machine translation systems. The problem is the difficulty of establishing a concrete mapping of the true semantic-syntactic structure of a sentence to its internal logical representation, which is automatically generated by the system [52-57].

These types of ambiguities can be resolved by introducing additional values that will increase the programmer's knowledge of a particular industry. Today, there are no programs that "understand" all types of ambiguities in a wide range of industries, but there are programs that can correctly respond to ambiguities in very narrow areas

The Main Tasks of Natural Language Processing

1. Data mining: study data, search for relationships and patterns between them.
2. Speech Synthesis: Speaking / reading text with a voice that is close to natural.
3. Language recognition: output / recognize text from pictures, scanned documents, or files in PDF format. This includes speech recognition produced by the human voice.
4. Natural language generation: converting computer data into human natural language.
5. Machine Translation: Automatic translation from one human language to another. This task is extremely difficult, because the machine does not have the knowledge that the person has, which makes them "understand" certain phrases completely different.
6. Question and Answer Systems: Answers to human-language questions. Usually the questions are specific, such as "Where is the Eiffel Tower?" But there are

questions that do not have a specific answer, such as “Why are all people different?” which makes this task extremely difficult to accomplish.
7. Topic Recognition / Definition: Divide the text into parts, then define the leading theme for each.
8. Information search: search, identify and retrieve information.
9. Data Retrieval: Retrieve semantic information from text.
10. Getting Connected: Defining relationships between objects in a specific piece of text (for example, who works with whom).
11. Text Simplification: Modifying, extending, or otherwise processing information to simplify the structure or grammar of the text while retaining the basic idea.
12. Lexicon Resolution: Provide a list of possible meanings of a particular multi-valued word, among which you can choose the most appropriate one in context.
13. Acronym and title recognition.
14. Detection of individual linguistic units.
15. Morphological decomposition: converting individual terms (such as medical or technical ones) into a comprehensible form.

Approaches to Natural Language Processing Tasks

Statistical Approach

The statistical approach to natural language processing is based on the assumption that the content of the text can be determined by the most common words. The main objective of this approach is to determine the number of repetitions of a particular word in the text. The latent-semantic approach is a variant of the statistical method and is based on the idea that the totality of all contexts in which a given word occurs or does not occur determines many mutual constraints for identifying similarities in word meanings. The main problem facing statistical approaches is to consider the text as a set of words without a meaningful connection [58].

Linguistic Approach

The linguistic approach to natural language processing consists of four levels: graphematic, morphological, syntactic and semantic [6]. The first level is to identify the individual elements of the text / document, such as sections, paragraphs, sentences, etc. The second level is to determine the morphological characteristics of each word. The third level is responsible for determining the syntactic dependence of words in sentences. The last level is related to the semantic understanding of the text, including developments in the field of artificial intelligence. Research achievements in this field are very limited due to the complexity of human language [59].

Symbolic Approach

The symbolic approach to natural language processing performs an in-depth analysis of linguistic phenomena and is based on the explicit representation of knowledge through the use of well-researched knowledge representation schemes and algorithms that work with them [7]. Dictionaries, formulas, and rules developed by humans can be the source of language knowledge [60].

The Connecticut Approach

This natural language processing method is responsible for processing common models using specific examples of language phenomena. The most significant difference between the Connecticut approach and other statistical methods is the combination of statistical knowledge and different theories of ideas that allow us to work with logical inferences and transformation of logical formulas [61].

Method of Auxiliary Vectors

Differential machine learning method that helps classifies words into categories. This method is based on a set of properties [62].

Hidden Markov Model

This is a graphical system in which each vertex is a random variable that can acquire any value (with certain probabilities) between several states, producing one of several possible source characters with each transition. The set of all possible states and unique symbols can be large. We can see the source data, but the initial states of the system are hidden [63].

Conditional Random Fields

Separate (differential) model that generates logistic regression for the data sequence. Used to predict the state of a variable is based on the observed variable [64].

N-Gram Models

The model is built on a sequence of n elements: sentences, words, letters, sounds, etc. The model allows you to calculate the probability of occurrence of any element under the known probabilities of occurrence of such previous elements. This model is reduced to a finite set of probabilities, each of which can be estimated after calculating the repetition of the corresponding n-grams [65].

Text Mining

Intelligent Text Analysis (ITA) is a field of data mining and artificial intelligence aimed at obtaining information from text document collections based on the application of effective, in a practical way, machine learning and natural language processing techniques. Text mining uses all the same approaches to processing information as data mining, but the difference between these areas is only in the final methods, and that data mining deals with repositories and databases, not electronic libraries and corpora of texts. The key tasks of the ITA are: categorizing texts, finding information, processing changes in text collections, and developing tools for presenting information to the user. Document categorization consists of collating documents from a collection with one or more groups (classes, clusters) of similar texts (for example, by theme or style). Categorization can take place with or without human involvement.

In the first case (document classification), the ITA system must assign the texts to the classes already defined (convenient for her). This requires training with the teacher, for which the user must provide the ITA system with both a list of classes and sample documents belonging to these classes [67-70].

The second case of categorization is called document clustering. In this case, the ITA system must itself determine the number of clusters by which texts can be distributed - in machine learning, the corresponding task is called learning without a teacher. In this case, the user must tell the ITA system the number of clusters that he would like to split the processed collection (assuming that the algorithm of the program already has a procedure for selecting features)

3 Statement of the Problem of Creating the Identification and Generation System Semantically Similar Sentences

3.1 Semantic Similarity and Paraphrase

Microsoft Research Paraphrase Corpus Dataset

The Microsoft Research Paraphrase Corpus (MSRP) consists of 5801 pairs of sentences, each accompanied by a binary judgment indicating whether human raters considered the pair of sentences to be similar enough in meaning to be considered close paraphrases. This data has been published for the purpose of encouraging research in areas relating to paraphrase and sentential synonymy and inference, and to establish a discourse on the proper construction of paraphrase corpora for training and evaluation. Paraphrasing can be seen as a logical inference on both sides. In this data set, one sentence has information that is not otherwise worded, that is, it does not have a strict paraphrase, and a degree of freedom is allowed.

Quora Question Pairs Dataset

Our dataset consists of over 400,000 lines of potential question duplicate pairs. Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair. Here are a few sample lines of the dataset:

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Fig. 5. Sample Data Quora Question Pairs

Here are a few important things about this dataset:

- Our original sampling method returned an imbalanced dataset with many more true examples of duplicate pairs than non-duplicates. Therefore, we supplemented the dataset with negative examples.
- The distribution of questions in the dataset should not be taken to be representative of the distribution of questions asked on Quora.

- The ground-truth labels contain some amount of noise: they are not guaranteed to be perfect.

3.2 Semantic Similarity and Paraphrase

MultiNLI

The Multi-Genre Natural Language Inference (MultiNLI) dataset is a dataset of 433,000 examples and is the largest logical inference recognition tool. MultiNLI includes ten different genres of written and spoken English, making it possible to test systems in a language close to the real complexity of the language.

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11 **entailment** E E E E	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE **neutral** N N E N	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE **contradiction** C C C C	No one noticed and it wasn't funny at all.

Fig. 6. Random selected examples from MultiNLI

3.3 The General Statement of the Problem of Identification of Semantically Similar Sentences

Natural language sentences should be classified into 3 classes: "entailment", "neutral", "contradiction". “Entailment” - the meaning of the second sentence can be logically deduced from the first. “Neutral” is the meaning of the second sentence cannot be logically deduced from the first, because of insufficient data in the first sentence, or because of the different content of the sentences (e.g. sentences from different domains). “Contradiction” - the meaning of the second sentence contradicts the meaning of the first.

3.4 The General Statement of the Problem of Generating Semantically Similar ones Sentence

The sentence of the sample is transformed into a sentence with a different wording but with the same meaning as the first. The semantic similarity condition must be fulfilled between the two sentences. That is, when classifying this pair of sentences, they must be classified as “entailment”.

3.5 Specification of Software Requirements

The purpose of this project is to create a system for automatically identifying and generating semantically similar sentences.

The following main characteristics can be distinguished:

Identifying paraphrased sentences that do not or almost do not have the same words. Flexibility to recognize sentences by changing the structure of the sentence is using synonyms or antonyms.

There will only be one class in the system: application users. This app is for people who need to automatically identify or generate semantically similar sentences.

Description and Priorit

1. The priority is medium. Ability save the results of generating and identifying semantically similar sentences.
2. The action-response sequence.
   (a) The user opens the application.
   (b) The user specifies the corresponding parameter in the console and the file name in which the results will be stored.
3. Functional requirements.
   (a) REQ 1. Informative message that the save process is starting.
   (b) REQ 2. Enable saving.

Identification of Semantically Similar Sentences

4. Description and priority. The priority is high. The ability automatically identifies semantically similar sentences.
5. The action-response sequence.
   (a) The user opens the application.
   (b) The user specifies the sentence to be identified.
6. Functional requirements.
   (a) REQ 1. The accuracy of identification must be sufficiently high to make it more effective than non-automatic means.
   (b) REQ 2. Allow cancellation of the identification process.

Generation of Semantically Similar Sentences

1. Description and priority. The priority is high. The ability automatically generates semantically similar sentences.
2. The action-response sequence,
   (a) The user opens the application.
   (b) The user specifies the sentence to be paraphrased.
3. Functional requirements.
   (a) REQ 1. The precision of the paraphrase must be sufficiently high, sufficient to be more efficient than generating the paraphrase in a non-automatic manner.
   (b) REQ 2. Allow cancellation of the generation process.

Requirements for External Interfaces

4. User interfaces. The user can interact with the system using a personal computer that has enough computing resources to work with the system.
5. Hardware interfaces. The current system will not use any hardware interfaces.
6. Software interfaces: NLTK; PyTorch; Keras.

Other Non-Functional Requirements

1. Performance requirements. The system must quickly identify and generate sentences without having to search through large databases of ready-made samples.
2. Security requirements. Personal information is confidential and is not shared with third parties. This can be done by making this an open source system.
3. Quality attributes of the software product.
   (a) Ease of use.
   (b) Reliability.
   (c) Convenience of support.

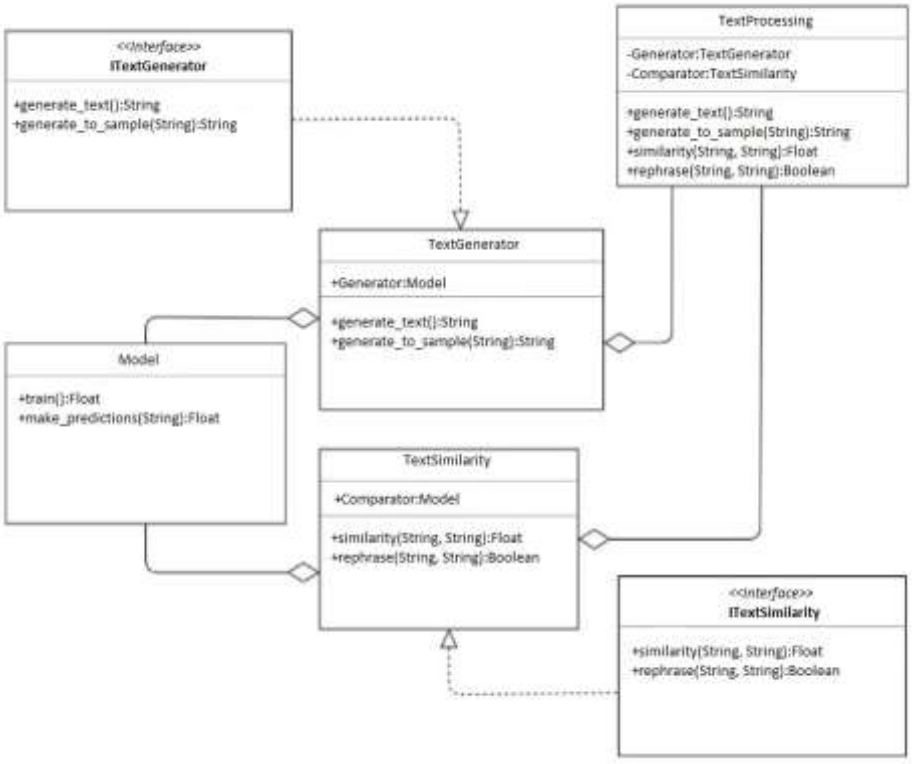

Fig. 7. Class diagram

3.6 Conceptual Model

Formulation of a meaningful and internal is view that combines the concept of user and model developer. It explicitly includes logic, algorithms, assumptions, and constraints. An abstract model is that reveals the causal relationships inherent in the object under study, within the limits defined by the objectives of the study. In essence, it is a formal description of a simulation object that reflects the researcher's concept (view) of the problem. Conceptual model is a domain model consisting of a list of interrelated concepts used to describe this field, together with the properties and characteristics, classification of these concepts, by types, situations, features in the field and the laws of the processes in it.

A conceptual (meaningful) model is an abstract model that defines the structure of a simulated system, the properties of its elements, and the causal relationships inherent in the system and essential to achieving the goal of modeling.

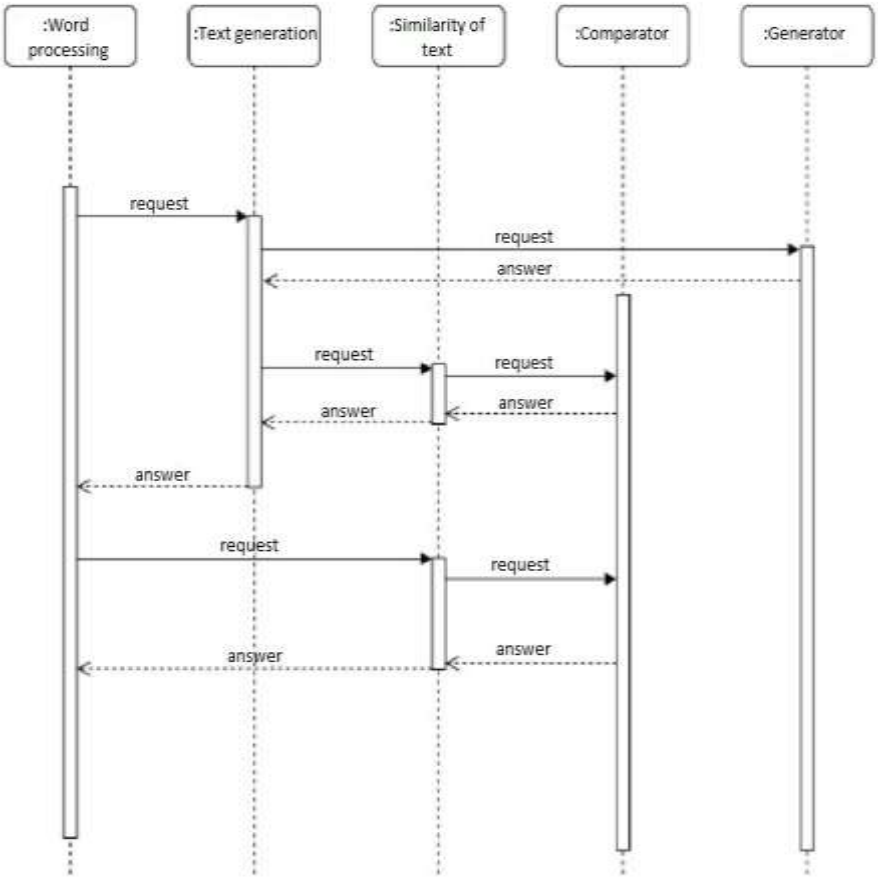

Fig. 8. Sequence diagram

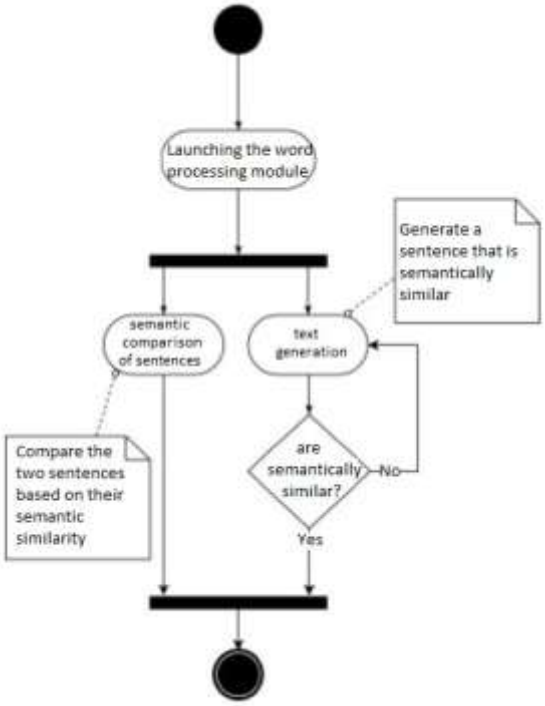

Fig. 9. Activity diagram

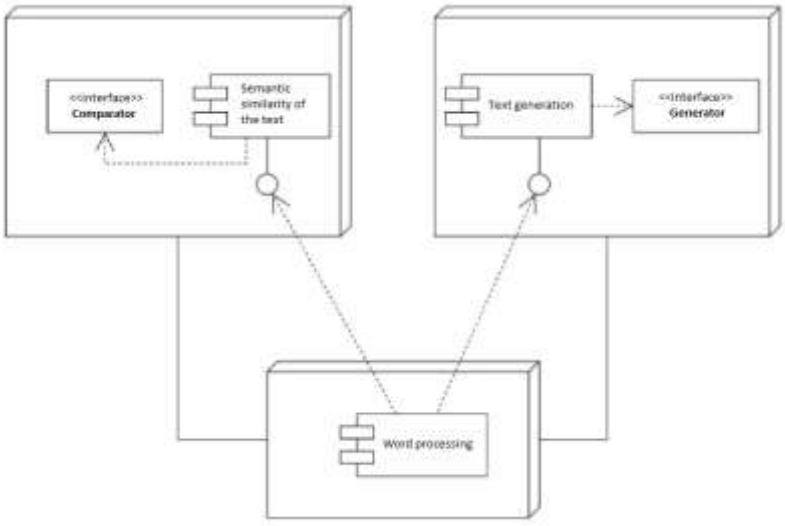

Fig. 10. Deployment diagram

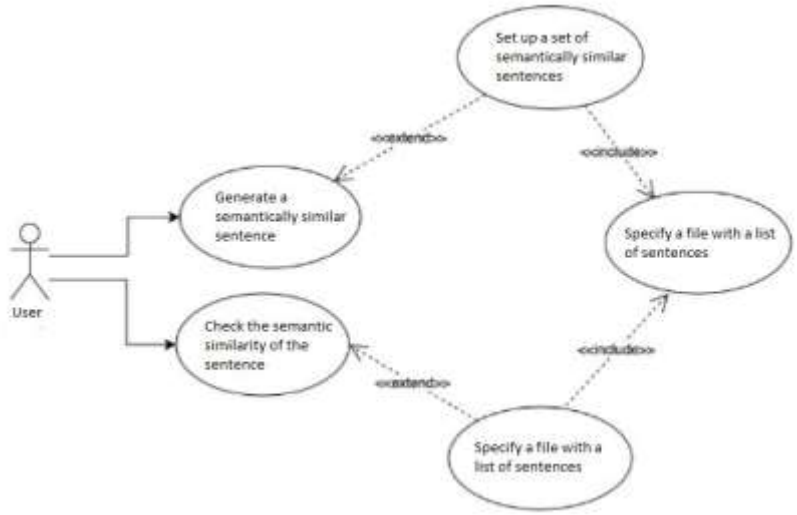

Fig. 11. Use case diagram

4 Design of Identification and Application Generation of Semantically Similar Sentences

4.1 Analysis of Approaches for Identification of Semantically Similar Sentences

The model Roberta shows state-of-the-art results on GLUE, RACE and SQuAD. This is a model that has the same architecture as the BERT model, but with minor modifications. It has been demonstrated that model accuracy improves significantly with model training longer with larger batches on more data; taking away the purpose of predicting the next sentence; by training on longer sequences; and consistently changing the mask pattern when training.

4.2 Analysis of Approaches for Generating Semantically Similar Sentences

GPT-2 Text Generation Model

Natural language tasks such as answering questions, summarizing, machine translation, and text comprehension are often accomplished by training with a teacher on specialized datasets. This model demonstrates that the model begins to learn these tasks without explicitly trained when it is trained on a data set of millions of web pages called WebText. An interesting feature of the model is that it is trained on language modeling tasks, although it shows some success in tasks that it has not been explicitly trained on. The largest GPT-2 model has 1.5 billion parameters and the

Transformer architecture. It achieves state of the art results in 7 of the 8 tested zero-shot language simulation datasets, but still has underfit on WebText.

Binary relation

Binary relation on a set is in mathematics a separate case of the relation given on a set M, which is established between two elements of the set. In other words, it is a subset of the Cartesian square $M2 = M \times M$. It is also said that the elements $a, b \in M$ are in the binary relation R (often written as aRb) if the ordered pair $(a, b) \in R$ and record that R as $M \times M$. In general, the binary relation between two sets A and B is a subset of $A \times B$. In this case, the term correspondence between sets is used. The term 2-digit ratio or 2-ratio is synonymous with binary ratio. In some systems of axioms of set theory, relations extend to classes, which are generalizations of sets. Such extension is needed, in particular, to formalize the notion of "being an element" or "being a subset" of set theory and preventing discrepancies such as Russell's paradox.

Types of relationships

- The reflexive transitive relation is called the quasi-order relation.
- The reflexive symmetric transitive relation is called the equivalence relation.
- A reflexive antisymmetric transitive relation is called a (partial) order.
- The anti-flexive antisymmetric transitive relation is called the strict order relation.
- Complete antisymmetric (for any x, y xRy or yRx holds) a transitive relation is called a linear order relation.
- The anti-flexural antisymmetric relation is called the dominance relation.

Since the relations on M are also sets, theoretically multiple operations are allowed over them. Example:

- The intersection of binary relations of "greater than or equal to" and "less than or equal to" is the ratio of "equal to"
- The union of "less" and "greater" is the ratio "not equal".
- The addition of the divisible by is the not divisible and the like.

The relation $R - 1$ is called inverted to the relation R if $bR - 1a$ if and only if aRb. Obviously, $(R - 1) - 1 = R$.

For example, for the ratio “greater than or equal to” the inverse is the relation “less than or equal”, for the relation “divisible by” - the relation “is divisor”.

Let R be some relation on the set M. The relation R is called:

- The inverse relation (the ratio opposite to R) is a binary relation consisting of pairs of elements (y, x) obtained by permutation of pairs of elements (x, y) of a given relation R. Denoted by: R-1. For this relation and its inverse it is true: (R-1)-1 = R.
- A reciprocal relationship is a relationship that is opposite to one another. The value area of one of them is the area of definition of the other, and the area of definition of the first is the area of values of the other.
- Reflexive if aRa holds for all $a \in M$.

- A binary relation R defined on some set, characterized in that for any x of this set the element x is relative to itself, that is, for any element x of that set xRx takes place.

Examples of reflexive relationships: equality, simultaneity, similarity. Relationship types:

- Antireflective (if not any a M $\in$ M does not hold aRa. Note that, just as antisymmetry does not coincide with asymmetry, irreflexivity does not coincide with non-reflexivity. The double relation R, defined on some set M, characterized in that for any element x of that set it is not fulfilled that it is relative to itself (absent xRx), that is, it is possible that the element of the set is not in relation R to himself. Examples of non-reflective relationships: "take care", "entertain", "nervous".
- Symmetric if for all a, b $\in$ M such that aRb we have bRa. The binary relation R, defined on some set, characterized in that for any elements of x and y of this set it follows that x is relative to R (xRy) and that y is in the same relation to x (uRx). An example of symmetric relations can be equality (=), the relation of equivalence, similarity, simultaneity, some relations of affinity.
- Asymmetric if for all a, b $\in$ M such that aRb does not hold bRa. The binary relation R, defined on some set, characterized in that for any x and y of xRy, the negation of yRx follows. Example: the ratio "greater" (>) and "less" (<).
- Antisymmetric if for all a, b $\in$ M such that aRb and a $\neq$ b, we have that bRa - does not hold. A binary relation R, defined on some set, characterized in that for any x and y with xRy and xR 1y, then x = y (ie, R and R − 1 are performed simultaneously only for equal terms).
- Transitive if aRc follows from the relations aRb and bRc. A binary relation R, defined on some set, characterized in that for any x, y, z of this set, xRy and yRz should be followed by xRz. Examples of transitive relations: "greater", "less", "equal", "like", "higher", "north".
- Non-transitive - binary relation R, defined on some set, characterized in that for any x, y, z of this set, xRy and yRz do not follow xRz. An example of a non-transitive relationship: "x father y".
- Complete if for any a, b $\in$ M it follows that aRb or bRa.
- Order relation - a relation having only some of the three properties of the equivalence relation. In particular, the relation reflexive and transitive but not symmetrical (for example, "no more") forms a "non-rigorous" order. The relationship is transitive but non-reflexive and asymmetric (for example, "less") - a "strict" order.
- Function - a binary relation R defined on some set, characterized in that each value x of the relation xRy corresponds to only one - a single value of y. Example: "y father x". The property of the functionality of the relation R is written as an axiom: (xRy and xRz) $\rightarrow$ (y $\equiv$ z). Since each value of x in the expressions xRy and xRz corresponds to the same value, then y and z will coincide, and will be the same. The functional relation is unambiguous, since to each value of x the relation xRy corresponds to only one - a single value of y, but not vice versa.
- Bijection is a binary relation R defined on some set, characterized in that in it each value of x corresponds to a single value of y, and to each value of y corresponds to a single value of x.

- Relationship relation is a binary relation R defined on some set, characterized in that for any two different elements x and y of this set, one of them is in relation to R to the other (ie one of two ratios: xRy or yRx). Example: less than (<). If the relation R has any of the above properties, then the inverse relation R − 1 also has the same property. Thus, the inversion operation retains all these relationship properties.

A relation that is reflexive, symmetrical, and transitive is called an equivalence relation. The notion of equivalence is closely related to the concept of partitioning.

Table 1. Title Properties

Name	reflexivity	anti-reflexivity	symmetry	asymmetry	anti-symmetry	transitivity	completeness
Advantage	+						
Similarity (tolerance)	+		+				
Equivalence	+		+			+	
Partially equivalence			+			+	
Quasi-order	+					+	
Ordering	+					+	+
Partial order	+				+	+	
Linear order	+				+	+	+
Austere quasi-order		+				+	
Austere order		+		+	(+)	+	
Domination		+		+	(+)		
Austere partial order		+		+	(+)	+	
Austere linear order		+		+	(+)	+	+

Equivalence Ratio

The equivalence ratio ($\approx$) on the set X is a binary relation for which the following conditions are satisfied: Reflexivity, Symmetry, Transitivity.

An entry of the form "a $\approx$ b" is read as "a is equivalent to b".

The consequence of the properties of reflexivity, symmetry and transitivity is that any equivalence relation provides for the division of any base set into disjoint equivalence classes. Two elements of a given set are equivalent if and only if they belong to the same class of equivalence. Examples of equivalence relations are

- The most striking example of equivalence is the division of students into classes.
- Equality relation is a trivial equivalence relation on an arbitrary set, in particular on the set of real numbers.
- Module comparison.

- In Euclidean geometry the relation of congruence, similarity and parallelism of straight lines.
- The ratio of equality of sets is the relation of equivalence.

Ways to Set Relationships

In order to specify the relation (R, Ω), it is necessary to specify all pairs of elements (x, y) ∈ Ω × Ω, which are included in the set R. In addition to the complete list of all pairs, there are three ways of defining relations: by means of a matrix, graph and cuts. The first two methods are used to define the relation on finite sets, the definition of the relation by sections can be applied to infinite sets.

Defining a Relation Using a Matrix

Let the set Ω consist of n elements, let R be the binary relation represented on that set. We number the elements of the set ми by integers from 1 to n. To define a relation, we construct a square table of size n × n. Its i-th row corresponds to the element xi of the set Ω, its jth column corresponds to the element xj of the set Ω. At the intersection of the i-th row and the j-th column, we set 1 if the element xi is in relation to R with the element xj, and zero in other cases.

Specify a Relation Using a Graph

In order to define a relation by means of a graph, we put in a one-to-one correspondence to the elements of the finite set Ω on which the relation is defined, the vertices of the graph x1, ..., xn (by any numbering).

It is possible to draw an arc from vertex xi to xj if and only if element xi is in relation to R with element xj, and if i = j, then the arc (xi, xj) becomes a loop at vertex xi.

Specify Relationships Using Cuts

The upper section of the relation (R, Ω) in the element x, denoted by R + (x), is the set of elements y ∈ Ω, for which the condition is satisfied: (y, x)∈R, R + (x) = {y ∈ Ω | (y, x) ∈ R}. The lower section of the relation (R, Ω) in the element x, denoted by R - (x), is the set of elements y ∈ Ω, for which the condition is satisfied: (x, y) ∈ R, namely R - (x) = {y ∈ Ω | (x, y) ∈ R}.

Therefore, the upper section (set R +) is the set of all such elements y that are in relation to R with a fixed element x (yRx). The lower section (the set R-) is the set of all such elements y with which the fixed element x is in relation to R (xRy).

So, in order to define a relation by means of cuts, it is necessary to describe all its upper or all lower sections. That is, the relation R will be given if a set R + (x) is given for each element x ∈ or a set R - (x) is given for each element x ∈ Ω.

4.3 Analysis of the Transitivity of the Recognising Textual Entailment Task

If for the Recognising Textual Entailment task the transitivity relations contained between the expression set are executed, then they can be represented in a hierarchical graph structure. [17]

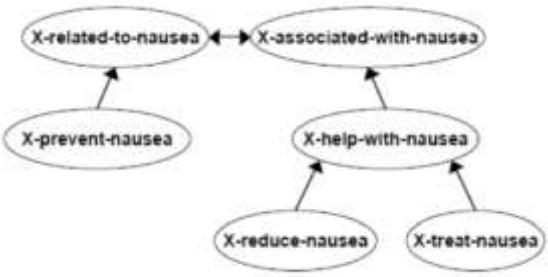

Fig. 12. Hierarchical graph structure of transitive dependencies between words. [17]

The data presented as such structure can be used to train the model. The model is trained on data in which there are transitive relations will learn to implement this relationship.

4.4 System Design Choices

The mouse look recognition system will be composed of two components to control the mouse pointer:

1. SemanticIdf is which will be able to identify semantically similar sentences.
2. SemanticGen is a system for generating semantically similar sentences.

Train a machine learning model on a large set of pairs of sentences. From each pair of sentences extract features such as modal verbs, numerical values, Levenstein distance and others. Check that the model satisfies similarity conditions such as reflexivity, symmetry, and transitivity.

Generate sentences with a pre-trained model, with the goal of generating the most semantically similar sentences. The algorithm can be reduced to generating sentences and verifying that the sentence is semantically similar to the sample.

5 Describing the Innovation of the Task

The aim of the thesis is to implement an intellectual system of identification and generation of semantically similar sentences based on machine learning methods.

An innovative component of this thesis is the study of transitivity in the RTE problem, because existing models do not explicitly implement this relation. That is, if one sentence logically follows the second and the third, then the logical derivation of the third sentence from the first should also be performed. That is, if $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$. must also be performed. Also, the scope of the thesis work is to study the limitations of existing datasets to solve the RTE problem, and the constraints on solving this problem as such in terms of philosophy.

6 Analysis of the Results

The MultiNLI dataset was used to train and test the system. From it, a subset of a dataset of one hundred thousand examples was selected. Twenty-five percent of the data was earmarked for system testing, and the rest was earmarked for training.

The cosine of similarity

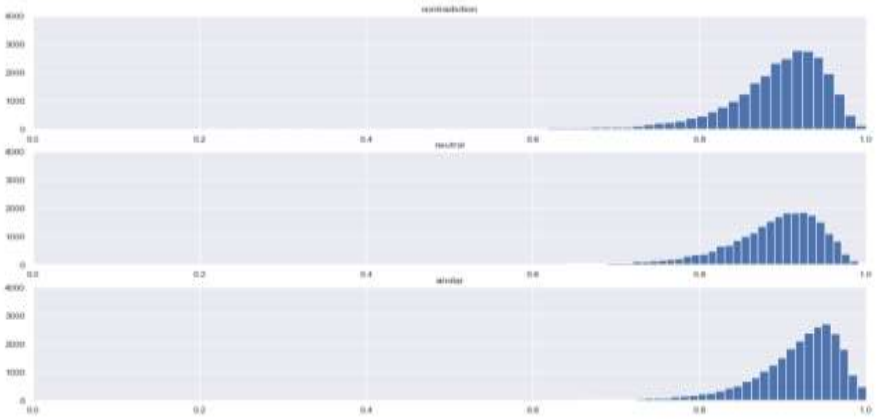

Fig. 13. Histograms for the cosines of similarity for each of the classes

The arithmetic mean for the classes "similar", "neutral" and "contradiction" is 0.914, 0.885 and 0.89 respectively. The median for "similar", "neutral" and "contradiction" is 0.928, 0.898 and 0.904 respectively. The mean square deviation is 0.062, 0.068, and 0.07, respectively. We can see that although the highest arithmetic mean and median are in the class “similar”, however, the difference between the values of the different classes is not large enough to be easily classified using this sign. Considering the distributions of different classes, we can conclude that these classes are not clearly separable using only the values of the cosine of similarity between vectors.

We use the cosine of the angle between vectors to train a linear classification model. The results of the model verification are presented in Table 2.

Table 2. The value of logistic regression accuracy metrics for each class.

Metric and class name	The value of the metric
Accuracy	0.40497333333333335
Precision “similar”	0.42551798203106583
Precision “neutral”	0.37994034302759133
Precision “contradiction”	0.3839664919012331
Recall “similar”	0.6424503677924543
Recall “neutral”	0.043975487657517694
Recall “contradiction”	0.4938964659784493

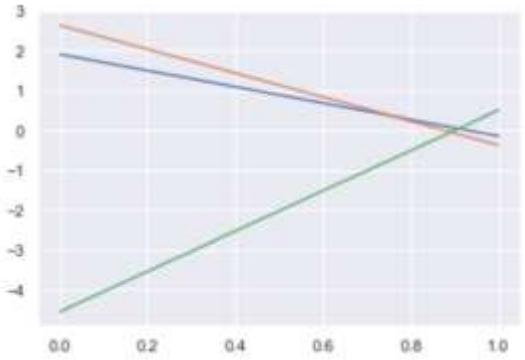

Fig. 14. The lines that were constructed by logistic regression to separate each of the classes

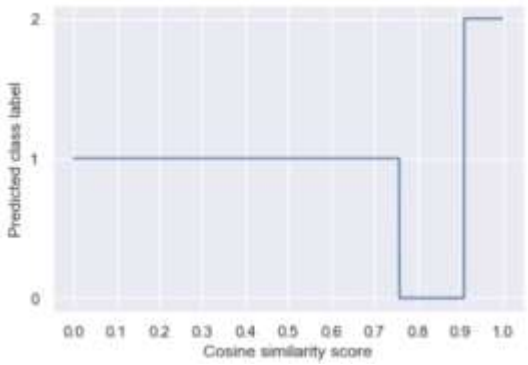

Fig. 15. Graph of the dependence of the cosine of similarity and the result of logistic regression prediction

From graph 4.3 we can see that logistic regression classifies a pair of sentences as "neutral" if the cosine of similarity between the vectors of this pair of sentences lies in the intermediate range from 0 to 0.75. At values of cosine of similarity from 0.75 to 0.9, the model produces a result of "contradiction" and at values from 0.9 to 1 - "similar". The fact that the model for the class "contradiction" uses greater values of cosine similarity than for the class "neutral" indicates that "contradiction" has a formulation of pain similar to "similar" than "neutral", which is not a good result because "contradiction "Should have smaller values of similarity cosine than "neutral".

Table 3. The metrics values

Metric and class name	The value of the metric	
	SGD classifier metric accuracy metrics for each class	the accuracy of the reference vector method for each class
Accuracy	0.3522533333333333	0.40784
Precision "similar"	0.3486562736315514	0.43468647238915975
Precision "neutral"	0.22058823529411764	0

Precision “contradiction”	0.39380674448767833	0.38399959722082366
Recall “similar”	0.9481531282132405	0.6064225262991378
Recall “neutral”	0.000647332988089073	0
Recall “contradiction”	0.09151533418732574	0.5747117775600934

6.1 The Average Between a Couple of Sentence Embedding

Table 4. The values of logistic regression accuracy metrics for each class when using the average between sentences of a couple of sentences

Metric and class name	The value of the metric
Accuracy	0.50164
Precision “similar”	0.48400272294077606
Precision “neutral”	0.4658757850662945
Precision “contradiction”	0.5472785722203747
Recall “similar”	0.5031847133757962
Recall “neutral”	0.42971163748712665
Recall “contradiction”	0.5639707562257253

Table 5. The Mean Between the Couple Word Embryos and the Cosine of Similarity Between the Word Pair Vectors

Metric and class name	The value of the metric	
	logistic regression accuracy metrics for each class using the mean between embedies and the cosine of similarity between pairs of sentences	Random Forest accuracy metric values for each class using medium between embedding and cosine similarity between pairs of sentences
Accuracy	0.54196	0.47148
Precision “similar”	0.5365727310401989	0.4736957474791758
Precision “neutral”	0.5303206997084549	0.48207101626727306
Precision “contradiction”	0.557492931196984	0.46352987498769566
Recall “similar”	0.6108752064166076	0.5097900448218919
Recall “neutral”	0.46833161688980435	0.354788877445932
Recall “contradiction”	0.5405528901073795	0.5379255197623943

6.2 Character Distance Between Pairs of Sentences

Table 6. The values of logistic regression accuracy metrics for each class using the symbolic distance between pairs of sentences

Metric and class name	The value of the metric
Accuracy	0.3724
Precision “similar”	0.7134606317774634
Precision “neutral”	0.002957121734844751
Precision “contradiction”	0.3848809523809524
Recall “similar”	0.356835465424748
Recall “neutral”	0.26666666666666666
Recall “contradiction”	0.40682018371712597

6.3 The Intersection of Words Between Pairs of Sentences

Table 7. The values of logistic regression accuracy metrics for each class when using the word-by-word crossing between pairs of sentences

Metric and class name	The value of the metric
Accuracy	0.40008
Precision “similar”	0.7511786892975012
Precision “neutral”	0
Precision “contradiction”	0.43202380952380953
Recall “similar”	0.367680147695148
Recall “neutral”	0
Recall “contradiction”	0.47332724664145037

6.4 The Length of the Sentence as a Sign for Classification

Table 8. The value of logistic regression accuracy metrics for each class when using sentence lengths

Metric and class name	The value of the metric
Accuracy	0.37364
Precision “similar”	0.3248443689869836
Precision “neutral”	0.40391943385955364
Precision “contradiction”	0.37398934503290504
Recall “similar”	0.13666666666666666
Recall “neutral”	0.2742730409068507
Recall “contradiction”	0.7033239038189534

6.5 Number of Words as a Sign for Classification

Table 9. The value of logistic regression accuracy metrics for each class using word count

Metric and class name	The value of the metric
Accuracy	0.37472
Precision “similar”	0.32454212454212455
Precision “neutral”	0.40823844608171467
Precision “contradiction”	0.3708430482267763
Recall “similar”	0.10547619047619047
Recall “neutral”	0.3003942828979793
Recall “contradiction”	0.7123998114097124

6.6 Simple Classifiers

Table 10. Values of logistic regression accuracy metrics using simple classifiers

The name of the classifier	The accuracy value
Most common class classifier	0.33936
Stratified classifier	0.33712

Table 11. Combine All the Features Together

Metric and class name	The value of the metric	
	The value of logistic regression accuracy metrics for each class when all traits are combined	Random Forest accuracy metric values for each class when combining all features together
Accuracy	0.56468	0.49868
Precision “similar”	0.5648089508127507	0.499311075781664
Precision “neutral”	0.5616968357054027	0.5236065573770492
Precision “contradiction”	0.567137169743033	0.481986265187533
Recall “similar”	0.6311630101439019	0.5556735079028072
Recall “neutral”	0.5233007209062822	0.4111740473738414
Recall “contradiction”	0.5370116518163125	0.5211331962531415

7 Conclusion

During the analytical review of literary and other sources, the problems of identification and generation of semantically similar sentences were analyzed. Areas of application of identification and generation of semantically similar sentences were ana-

lyzed. In this article, the formulation of the problem was performed and the datasets used to solve the problem were analyzed. The specification of requirements is made. Describes the ways and means to develop a user-recognition application to control the mouse pointer, and outlines the benefits of the chosen direction for solving the tasks involved in developing the application. Analyzing the types of binary relations, we can conclude that the relationship between pairs of sentences in the task of Recognizing Textual Entailment is reflective and transitive.

This article analyzes the approaches to solving the problems of identifying and generating semantically similar sentences. This article describes all the main technical aspects of solving this problem, describes proposed solutions for the development of algorithmic, functional and software components of the application of identification and generation of semantically similar sentences. During the analysis of existing algorithms, the basic principles of operation of such algorithms were considered. Analogues were analyzed, namely the methods of semantic comparison of sentences, their advantages and disadvantages were determined. The methods that solve the problem are many, but they have some limitations, such as unreliability after slight changes to the text or paraphrase. This article describes the software implementation of the task. Different ways of semantic comparison and text generation are analyzed. Also, the system was tested for new data, that is, data that was not used to train the model.

References

1. Google AI blog, https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html, last accessed 2020/02/21.
2. Learning Semantic Textual Similarity from Conversations, https://uk.wikipedia.org/wiki/, last accessed 2020/02/21.
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546 [cs.CL]. (2013).
4. Lebret, R., Collobert, R.: Word Emdeddings through Hellinger PCA. In: Conference of the European Chapter of the Association for Computational Linguistics (EACL). arXiv:1312.5542. Bibcode:2013arXiv1312.5542L. (2013).
5. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. NIPS. (2014).
6. Li, Y., Xu, L.: Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective. In: Int. J. Conf. on Artificial Intelligence (IJCAI). (2015).
7. Globerson, A.: Euclidean Embedding of Co-occurrence Data. In: Journal of Machine Learning Research. (2007).
8. Qureshi, M. A., Greene, D.: EVE: explainable vector based embedding technique using Wikipedia. In: Journal of Intelligent Information Systems. arXiv:1702.06891. (2018).
9. Levy, O., Goldberg, Y.: Linguistic Regularities in Sparse and Explicit Word Representations. In: CoNLL, 171–180. (2014).
10. Towson University “Sentence patterns”, https://webapps.towson.edu/ows/sentpatt.htm, last accessed 2016/11/21.
11. Stanford Natural Language Processing Group “Neural Network Dependency Parser”, https:// nlp.stanford.edu/software/nndep.html, last accessed 2016/11/21.
12. Edit Distance and Postediting, https://www.gala-global.org/blog/edit-distance-and-postediting, last accessed 2016/11/21.

13. Nlp Town “Comparing Sentence Similarity Methods”, http://nlp.town/blog/sentence-similarity, last accessed 2016/11/21.
14. ClearNLP Dependency Labels, https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md, last accessed 2016/11/21.
15. TEG-REP: A Corpus of Textual Entailment Graphs based on Relation Extraction Patterns, https://www.researchgate.net/publication/297759246_TEG-REP_A_Corpus_of_Textual_Entailment_Graphs_based_on_Relation_Extraction_Patterns, last accessed 2016/11/21.
16. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015, 120-123 (2015)
17. Korobchinsky, M., Vysotska, V., Chyrun, L., Chyrun, L.: Peculiarities of Content Forming and Analysis in Internet Newspaper Covering Music News, In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 52-57 (2017)
18. Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., Rishnyak, I.: The Contextual Search Method Based on Domain Thesaurus. In: Advances in Intelligent Systems and Computing, 689, 310-319 (2018)
19. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216 (2018)
20. Naum, O., Chyrun, L., Kanishcheva, O., Vysotska, V.: Intellectual System Design for Content Formation. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 131-138 (2017)
21. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 2, 113-116 (2018)
22. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city, CEUR Workshop Proceedings, 239-255 (2018)
23. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 343-346 (2018)
24. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies, Advances in Intelligent Systems and Computing, 871, 206-221 (2019)
25. Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., Burov, Y.: Information resources processing using linguistic analysis of textual content. In: Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications, Romania, 573-578, (2017)
26. Vysotska, V.: Linguistic Analysis of Textual Commercial Content for Information Resources Processing. In: Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET’2016, 709–713 (2016)
27. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 124-128 (2015)
28. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136 (2018)

29. Vasyl, Lytvyn, Victoria, Vysotska, Dmytro, Dosyn, Roman, Holoschuk, Zoriana, Rybchak: Application of Sentence Parsing for Determining Keywords in Ukrainian Texts. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 326-331 (2017)
30. Lytvyn, V., Vysotska, V., Uhryn, D., Hrendus, M., Naum, O.: Analysis of statistical methods for stable combinations determination of keywords identification. In: Eastern-European Journal of Enterprise Technologies, 2/2(92), 23-37 (2018)
31. Chyrun, L., Vysotska, V., Kis, I., Chyrun, L.: Content Analysis Method for Cut Formation of Human Psychological State, Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, 139-144 (2018)
32. Vysotska, V., Chyrun, L., Chyrun, L.: Information Technology of Processing Information Resources in Electronic Content Commerce Systems. In: Computer Science and Information Technologies, CSIT'2016, 212-222 (2016)
33. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V., Kovalchuk, R., Petruchenko, O., Dzyubyk, L., Sokulska, N.: Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. In: Eastern-European Journal of Enterprise Technologies, 2(2-98), 15-34. (2019)
34. Chyrun, L., Kis, I., Vysotska, V., Chyrun, L.: Content monitoring method for cut formation of person psychological state in social scoring. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, 106-112 (2018)
35. Vysotska, V., Lytvyn, V., Burov, Y., Berezin, P., Emmerich, M., Fernandes, V. B.: Development of Information System for Textual Content Categorizing Based on Ontology. In: CEUR Workshop Proceedings, Vol-2362, 53-70 (2019)
36. Lytvyn, V., Pukach, P., Bobyk, I., Vysotska, V.: The method of formation of the status of personality understanding based on the content analysis. In: Eastern-European Journal of Enterprise Technologies, 5/2(83), 4-12 (2016)
37. Lytvyn V., Vysotska V., Pukach P., Nytrebych Z., Demkiv I., Kovalchuk R., Huzyk N.: Development of the linguometric method for automatic identification of the author of text content based on statistical analysis of language diversity coefficients, Eastern-European Journal of Enterprise Technologies, 5(2), 16-28 (2018)
38. Lytvyn, V., Vysotska, V., Rzheuskyi, A.: Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. In: CEUR Workshop Proceedings, Vol-2392, 147-171. (2019)
39. Lytvyn, V., Vysotska, V., Rusyn, B., Pohreliuk, L., Berezin, P., Naum O.: Textual Content Categorizing Technology Development Based on Ontology. In: CEUR Workshop Proceedings, Vol-2386, 234-254. (2019)
40. Vysotska, V., Kanishcheva, O., Hlavcheva, Y.: Authorship Identification of the Scientific Text in Ukrainian with Using the Lingvometry Methods, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 2, 34-38 (2018)
41. Demchuk, A., Lytvyn, V., Vysotska, V., Dilai, M.: Methods and Means of Web Content Personalization for Commercial Information Products Distribution. In: Advances in Intelligent Systems and Computing, 1020, 332–347. (2020)
42. Vysotska, V., Lytvyn, V., Hrendus, M., Kubinska, S., Brodyak, O.: Method of textual information authorship analysis based on stylometry. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 9-16 (2018)

43. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O.: Automated Monitoring of Changes in Web Resources. In: Advances in Intelligent Systems and Computing, 1020, 348–363. (2020)
44. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Senyk, A., Malanchuk, O., Sachenko, S., Kovalchuk, R., Huzyk, N.: Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian, Eastern-European Journal of Enterprise Technologies, 6(2-96), pp. 19-31 (2018)
45. Vysotska, V., Fernandes, V.B., Lytvyn, V., Emmerich, M., Hrendus, M.: Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship, Advances in Intelligent Systems and Computing, 871, 132-151 (2019)
46. Vysotska, V., Burov, Y., Lytvyn, V., Demchuk, A.: Defining Author's Style for Plagiarism Detection in Academic Environment, Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, 128-133 (2018)
47. Lytvyn, V., Vysotska, V., Burov, Y., Bobyk, I., Ohirko, O.: The linguometric approach for co-authoring author's style definition, Proceedings of the 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2018, 29-34 (2018)
48. Lytvyn, V., Sharonova, N., Hamon, T., Cherednichenko, O., Grabar, N., Kowalska-Styczen, A., Vysotska, V.: Preface: Computational Linguistics and Intelligent Systems (COLINS-2019). In: CEUR Workshop Proceedings, Vol-2362. (2019)
49. Vysotska, V., Chyrun, L.: Methods of information resources processing in electronic content commerce systems. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM. (2015)
50. Andrunyk, V., Chyrun, L., Vysotska, V.: Electronic content commerce system development. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
51. Alieksieieva, K., Berko, A., Vysotska, V.: Technology of commercial web-resource processing. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
52. Lytvyn V., Vysotska V., Peleshchak I., Basyuk T., Kovalchuk V., Kubinska S., Chyrun L., Rusyn B., Pohreliuk L., Salo T.: Identifying Textual Content Based on Thematic Analysis of Similar Texts in Big Data. In: 2019 IEEE 14th International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT), 84-91. (2019)
53. Vysotska V., Lytvyn V., Kovalchuk V., Kubinska S., Dilai M., Rusyn B., Pohreliuk L., Chyrun L., Chyrun S., Brodyak O.: Method of Similar Textual Content Selection Based on Thematic Information Retrieval. In: 2019 IEEE 14th International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT), 1-6. (2019)
54. Khomytska, I., Teslyuk, V., Holovatyy, A., Morushko, O.: Development of methods, models, and means for the author attribution of a text. In: Eastern-European Journal of Enterprise Technologies, 3(2-93), 41–46. (2018)
55. Khomytska, I., Teslyuk, V.: Authorship and Style Attribution by Statistical Methods of Style Differentiation on the Phonological Level. In: Advances in Intelligent Systems and Computing III. AISC 871, Springer, 105–118, doi: 10.1007/978-3-030-01069-0_8 (2019)
56. Antonyu,k N., Medykovskyy, M., Chyrun, L., Dverii, M., Oborska, O., Krylyshyn, M., Vysotsky, A., Tsiura, N., Naum, O.: Online Tourism System Development for Searching and Planning Trips with User's Requirements. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, 1080, 831-863. (2020)

57. Lozynska, O., Savchuk, V., Pasichnyk, V.: Individual Sign Translator Component of Tourist Information System. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, Springer, Cham, 1080, 593-601. (2020)
58. Rzheuskyi, A., Kutyuk, O., Voloshyn, O., Kowalska-Styczen, A., Voloshyn, V., Chyrun, L., Chyrun, S., Peleshko, D., Rak, T.: The Intellectual System Development of Distant Competencies Analyzing for IT Recruitment. In: Advances in Intelligent Systems and Computing IV, Springer, Cham, 1080, 696-720. (2020)
59. Rusyn, B., Pohreliuk, L., Rzheuskyi, A., Kubik, R., Ryshkovets Y., Chyrun, L., Chyrun, S., Vysotskyi, A., Fernandes, V. B.: The Mobile Application Development Based on Online Music Library for Socializing in the World of Bard Songs and Scouts' Bonfires. In: Advances in Intelligent Systems and Computing IV, Springer, 1080, 734-756. (2020)
60. Sachenko, S., Rippa, S., Krupka, Y.: Pre-Conditions of Ontological Approaches Application for Knowledge Management in Accounting. In: IEEE International Workshop on Antelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Rende (Cozenza), Italy, 605-608. (2009)
61. Antonyuk N., Chyrun L., Andrunyk V., Vasevych A., Chyrun S., Gozhyj A., Kalinina I., Borzov Y.: Medical News Aggregation and Ranking of Taking into Account the User Needs. In: CEUR Workshop Proceedings, of the 2nd International Workshop on Informatics & Data-Driven Medicine (IDDM 2019), Vol-2362, 369-382. (2019)
62. Kis, Y., Chyrun, L., Tsymbaliak, T., Chyrun, L.: Development of System for Managers Relationship Management with Customers. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 405-421. (2020)
63. Chyrun, L., Kowalska-Styczen, A., Burov, Y., Berko, A., Vasevych, A., Pelekh, I., Ryshkovets, Y.: Heterogeneous Data with Agreed Content Aggregation System Development. In: CEUR Workshop Proceedings, Vol-2386, 35-54. (2019)
64. Chyrun, L., Burov, Y., Rusyn, B., Pohreliuk, L., Oleshek, O., Gozhyj, ., Bobyk, I.: Web Resource Changes Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2386, 255-273. (2019)
65. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O.: Uniform Method of Operative Content Management in Web Systems. In: CEUR Workshop Proceedings, Vol-2136, 62-77. (2018)
66. Chyrun, L., Gozhyj, A., Yevseyeva, I., Dosyn, D., Tyhonov, V., Zakharchuk, M.: Web Content Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2362, 126-142. (2019)
67. Kulchytskyi, I.: Statistical Analysis of the Short Stories by Roman Ivanychuk. In: CEUR Workshop Proceedings, Vol-2362, 312-321. (2019)
68. Shandruk, U.: Quantitative Characteristics of Key Words in Texts of Scientific Genre (on the Material of the Ukrainian Scientific Journal). In: CEUR Workshop Proceedings, Vol-2362, 163-172. (2019)
69. Levchenko, O., Romanyshyn, N., Dosyn, D.: Method of Automated Identification of Metaphoric Meaning in Adjective + Noun Word Combinations (Based on the Ukrainian Language). In: CEUR Workshop Proceedings, Vol-2386, 370-380. (2019)
70. Bisikalo, O., Ivanov, Y., Sholota, V.: Modeling the Phenomenological Concepts for Figurative Processing of Natural-Language Constructions. In: CEUR Workshop Proceedings, Vol-2362, 1-11. (2019)