

DOI [10.28925/2663-4023.2021.13.92101](https://doi.org/10.28925/2663-4023.2021.13.92101)

УДК 004.056.53

Кривенко Сергій Вікторович

д.т.н., доцент кафедри системного аналізу та інформаційних технологій
Маріупольський державний університет, м. Маріуполь, Україна
ORCID ID 0000-0002-0319-7174
s.krivenko@mdu.in.ua

Ротаньова Наталія Юріївна

к.пед.н., доцент кафедри системного аналізу та інформаційних технологій
Маріупольський державний університет, м. Маріуполь, Україна
ORCID ID 0000-0001-8437-7566
rotanovan@gmail.com

Лазаревська Юліанна Артурівна

асистент кафедри системного аналізу та інформаційних технологій
Маріупольський державний університет, м. Маріуполь, Україна
ORCID ID 0000-0001-8318-5861
lazarevskayulianna@gmail.com

АВТОМАТИЗОВАНА СИСТЕМА ВИЯВЛЕННЯ НЕСТАНДАРТНИХ ДІЙ ЗА ДОПОМОГОЮ СЦЕНАРНОГО АНАЛІЗА ТЕКСТУ

Анотація. Сценарій (narrative schemas) - деяка усталена (в соціумі) послідовність кроків для досягнення поставленої мети і містить максимально повну інформацію про всі можливі шляхи розвитку описуваної ситуації (з точками вибору та розгалуженнями). Створення XML-платформи поклало початок новому наукомісткому й технологічно більш досконалому етапу в розвитку Web. У результаті XML-платформа стає істотним компонентом у технології розробки інформаційних систем, а тенденція їх інтеграції на рівні корпорацій, відомств, міністерств тільки зміцнює позиції XML у сфері інформаційних технологій у цілому. Розроблено систему автоматичного виявлення нестандартних сценаріїв в текстових повідомленнях. Програмування системи складається з етапів формування онтологій, синтаксичного аналізу речення та порівняння сценаріїв. Для синтаксичного аналізу пропозицій в системі використаний класичний метод обробки природної мови (NLP), який підтримує найбільш поширені завдання, такі як токенизація, сегментація пропозицій, тегування частини мови, витяг іменованих сутностей, розбиття на частини, синтаксичний аналіз і дозвіл кореферентності. Також можлива максимальна ентропія і машинне навчання на основі перцептронів. Зберігання онтологій здійснюють за допомогою технології OWL. У процесі аналізу зіставляються парсери об'єкт-мета речення із описаною OWL. При виконанні з SPARQL запиту на об'єкті джерела або моделі запиту вертаються в об'єкт таблиці. Клас таблиці є базовим класом для всіх об'єктів таблиці й забезпечує інтерфейс для доступу до значень у рядках і стовпцях таблиці результатів. Якщо об'єкт таблиці має рівно три стовпці, то він може бути використаний для побудови нового об'єкта джерела даних. Це забезпечує зручний механізм для добування підмножини даних з одного джерела даних і додавання їх в іншій. У контексті RDF API вузол визначається як усі твердження про предмет URI. Вміст таблиці порівнюємо з семантикою речення. У разі не співпадіння сценарію речення з моделлю онтології OWL, існує вірогідність нетипових дій об'єкта. В такому варіанті формується висновок про підозрілість повідомлення. Для більш коректного використання можливостей аналізу тексту потрібно формувати корпус онтологій або використовувати існуючі (Akutan, Amazon та ін.) з врахуванням їх особливостей. Для збільшення онтологій об'єктів можливе додаткове використання нейромережевих методів навчання.

Ключові слова: онтологія; сценарій; парсер; аналіз змісту; семантичний аналіз



ВСТУП

Розвиток інтернет-мережі сприяє активному спілкуванню між людьми, зв'язку із громадськістю й прямого маркетингу, але також зв'язку дуже специфічної аудиторії, пов'язаною із протиправною діяльністю. Поступово із часом служби безпеки приходять до розуміння того, що потрібний автоматичний аналізатор текстів повідомлень, який без особливих затрат часу виконував би їхню оцінку з використанням методів теорії семантики.

Постановка проблеми. Багато із розробників інформаційно-аналітичних і пошукових систем заявляють про застосування методів семантичного аналізу. Однак, в ряді випадків семантичний аналіз підміняють інформаційними вимірюваннями тексту. Один з таких підходів полягає в обліку статистичних характеристик слів і їх сполучуваності, обліку семантичних класів слів. Ефективність використання таких підходів недостатня для вирішення завдань вилучення інформації, особливо при заміні основних понять неоднозначними за значенням.

Тому при обробці тексту необхідно застосовувати глибокий аналіз, об'єктом якого є кілька пропозицій, а не окрема дія (як у випадку з синтаксисом і семантикою). Для встановлення семантичних відносин між клаузами повинно використовувати дискурсивний аналіз, а для вилучення ланцюжків подій - сценарний аналіз. При цьому у кожного кроку є свої передумови (невідповідність яким не дозволить приступити до виконання кроку), дійові особи (яким пропонується виконати дану дію), а також операнди (об'єкти, з якими воно виконується). Тому він повинен формуватися на основі цілого корпусу, де кожен окремий текст містить певний сюжет, що описує особистий досвід автора по проходженню сценарію або його окремий випадок. Витяг сценаріїв забезпечить справжнє розуміння сенсу тексту.

Аналіз останніх досліджень і публікацій. Для розробки прикладних систем найчастіше застосовують комбіновані методи, коли низькорівневі завдання вирішуються машинним навчанням (витяг ознак, витяг фактів), а високорівневі (прикладні), завдання вирішуються системами правил [1-4]. Сценарний аналіз - наступний етап аналізу структури тексту, до результатів якого можна застосовувати правила або інші методи вирішення кінцевої завдання. У той час як методи, засновані на безперервних диференціюються моделях (наприклад, штучних нейронних мережах) показують гарну якість рішення деяких завдань, їх складно комбінувати з іншими підходами (наприклад, системами правил). До того ж, такі моделі будують векторні уявлення, які складно інтерпретувати («чорний ящик»). Тому часто процес аналізу тексту розбивають на декілька етапів (синтаксичний і семантичний розбір, витяг сутностей і фактів, зв'язування сутностей і т.п.), а потім додають правила, щоб на основі всього аналізу вирішувати кінцеве завдання - оцінювати якість документа, близькість текстів і т.д.

Більш універсальний когнітивний напрям пов'язаний з дослідженнями по штучному інтелекту (ШІ), який займається розробкою когнітивних процесуальних теорій, а також експериментами з комп'ютерного втілення цих теорій. Спираючись на процесуальний опис, ШІ моделює розуміння. Розуміння досягається об'єднаним застосуванням семантичних і прагматичних знань. Значення слова або вислова представлено як складова частина пам'яті, точно так же, як і інші знання. Найважливішим поняттям тут є «пам'ять» [5, 6].

Семантичні теорії повинні використовувати такі структури для представлення звичайних (common-sense) знань, як фрейми [7, 8] і сценарії (або скрипти - scripts) [9]. Кожне з цих слів несе в собі одночасно і фон, і зображення, і всю обстановку, і ту частину

цієї обстановки, на яку вказує дане слово. Причому перший використовувався в основному для подання статистичної інформації, наприклад, фрейм «кімната», а сценарій або скрипт - для процесуальної, що становить звичайну послідовність подій, наприклад, «відвідування ресторану». Скрипт - це деякий набір очікувань про те, що в сприймають ситуації має відбутися далі, усталена в культурі послідовність кроків, «знання того, як чинити і як інші надійдуть в конкретних стереотипних ситуаціях» [2].

Практично ці поняття стали використовуватися для генерації та аналізу текстів не в напрямку ШІ, зокрема, в генерації моделей таких стереотипних текстів, як, наприклад, анкета, біографія, зведення погоди. До цього ж типу можна віднести і більш спеціальні сценарії, наприклад, когнітивні схеми, які дозволяють лікарю запропонувати пацієнтові для обговорення сценарій його лікування.

Журавський і Чемберс в роботі [2] пропонують розглядати сценарій (narrative schemas) як послідовність взаємопов'язаних подій (наприклад, заарештувати, засудити), аргументи яких заповнені семантичними ролями учасників (поліція, суддя, обвинувачений). Даний підхід ґрунтується на понятті скрипта. Формально, сценарій пред-представляється у вигляді графа, в вершинах якого стоять дії разом з очікуваними аргументами, а ребра відповідають відношенню проходження. При цьому граф повинен містити всі можливі дії, які потенційно можуть виникнути в даній життєвій ситуації. Очевидно, такий граф неможливо побудувати по одному тексту, тому автори пропонують аналізувати колекцію текстів, присвячених одній ситуації.

Актуальність досліджень полягає у необхідності запобіганню використанню інтернет-мережі для координації протиправних дій та ще більше зростає в умовах проведення антитерористичної операції.

Мета статті. Розробка методики та системи автоматизованого сценарного аналізу тексту на ознаки протиправності змісту.

МЕТОДИКА ДОСЛІДЖЕННЯ

Аналіз сценарію тексту складається з послідовності дій, зображених на рис.1.

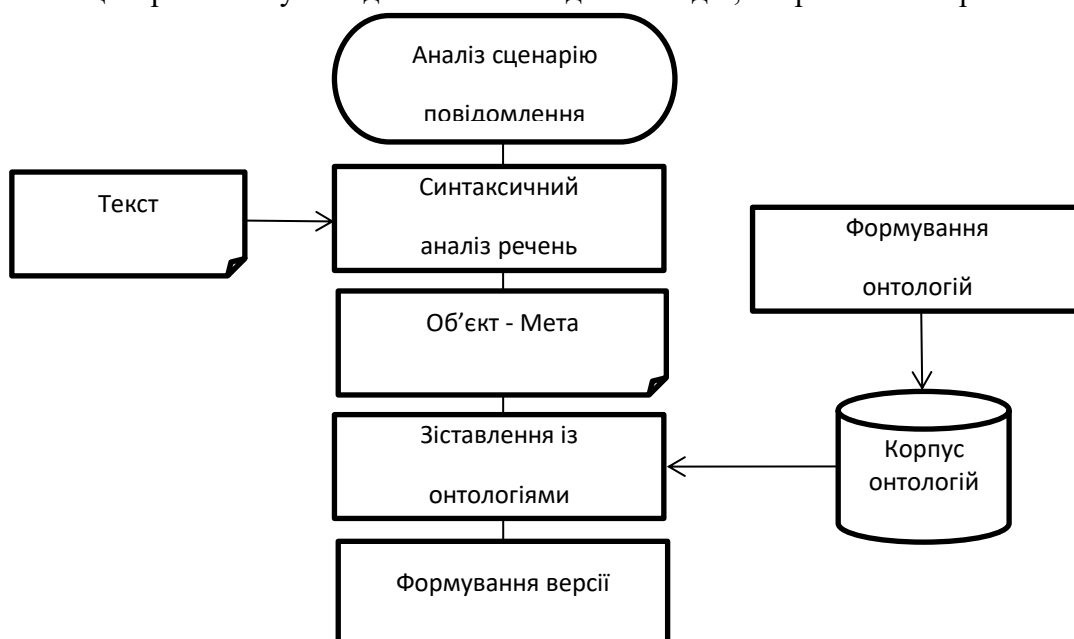


Рис. 1. Алгоритм аналізу сценарію повідомлення

Для синтаксичного аналізу пропозицій в системі використаний класичний метод обробки природної мови (NLP) [9-11]. Пакет Stanford NLP - це набір інструментів на основі машинного навчання для обробки тексту на природній мові. Він підтримує найбільш поширені завдання NLP, такі як токенізація, сегментація пропозицій, тегування частини мови, витяг іменованих сутностей, розбиття на частини, синтаксичний аналіз і дозвіл кореферентності. Stanford NLP також включає максимальну ентропію і машинне навчання на основі перцептронів.

Програмний модуль здійснює виділення токенів, аналіз настрою пропозиції, побудова дерева токенів із зазначенням частин мови, побудова бінарного дерева з анотаціями настроїв, аналіз залежностей, виділення ключових слів.

Методи класів, використані для аналізу тексту:

- опис аналізованого тексту

```
annotation = new Annotation("Prepare a loaf. Place it under the  
object.");
```
- опис дій над текстом

```
Properties props = new Properties();  
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner,  
parse, dcoref, sentiment");
```
- запуск всіх обраних анотаторів по тексту

```
pipeline.annotate(annotation);
```
- синтаксичний аналіз речення в тексті

```
List<CoreMap> sentences =  
annotation.get(CoreAnnotations.SentencesAnnotation.class);  
CoreMap sentence = sentences.get(0);  
sentence.get(CoreAnnotations.TokensAnnotation.class)  
Tree tree = sentence.get(TreeCoreAnnotations.TreeAnnotation.class);  
SemanticGraph graph =  
sentence.get(SemanticGraphCoreAnnotations.CollapsedCCProcessed-  
DependenciesAnnotation.class);
```

Для аналізу логіки сценарію необхідно сформувати базу знань предикатів виду Об'єкт <-> Мета, а також кореферентних зв'язку між ними [12, 13]. Для опису онтології використаний мову OWL. OWL (Web Ontology Language) - мова представлення онтологій наступного покоління після DAML + OIL. Володіє більш багатим набором можливостей, ніж XML, RDF, RDF Schema и DAML + OIL. Онтологія OWL - це послідовність аксіом, фактів і посилань на інші онтології.

Ієрархія класу ontology містить чотири підкласу (рис.2):

- action_verb - опис дій об'єкта.

Використана класифікація Г. Брінкманна [14], який розділив всі дієслова з того, що відбувається, на п'ять груп:

- 1) дії (позначають прямий вплив на об'єкт);
- 2) процес (передають зміни стану людей і предметів);
- 3) стан (висловлюють положення людей і предметів);
- 4) події;
- 5) явища природи.

- object_noun - опис об'єкта.

Крім назви об'єкта описані основні фізичні і хімічні властивості.

- description_adjective - ознаки об'єкта. Поділяються на якісні, відносні і присвійні.

Така ієрархія класів сприяє легшому її формування для кожного з об'єктів.

У розробленому модулі для роботи з онтологічними даними використана бібліотека класів OWL API:

```
import org.semanticweb.owlapi.search.Searcher.annotations;  
import org.semanticweb.owlapi.apibinding.OWLManager.*;  
import org.semanticweb.owlapi.vocab.OWLIRDFVocabulary.*;  
import org.semanticweb.owlapi.model.*;
```

Функціональність модуля полягає в:

- завантаженні й зберігання онтології OWL;
- запит даних OWL за допомогою SPARQL;
- виконання та збереження результатів по набору фактів.

Для реалізації доступного репозитора API OWL моделі забезпечений механізм керування джерелом даних у першу чергу для використання в запитах SPARQL-DL. Об'єкти джерела даних додаються в модель і пов'язуються з унікальним іменем у цій моделі. Ці джерела даних використовують по імені в запиті SPARQL [15]. Якщо в запит включено кілька джерел даних, запит залежить від усіх джерел даних. Результати будуть ідентичні тим, які були повернуті, якби всі дані перебували в одному джерелі даних. Модель також дозволяє додатку надавати користувацькі функції, які можуть бути включені в запит SPARQL або перевизначати реалізацію стандартних функцій.

```
// Create an ontology manager  
OWLManager manager = OWLManager.createOWLManager();  
// Load the ontology  
OWLManager ont =  
manager.loadOntologyFromOntologyDocument(IRI.create(""));
```

З бази онтологій зчитуються можливі сценарії за допомогою мови запитів SPARQL. Джерело даних виводу запитаний за допомогою SPARQL і підтримує той же інтерфейс, використовуваний джерелами даних без правил виводу. Основна відмінність полягає в тому, що результати запиту містять будь-які вивідні оператори, які відповідають умовам запиту. Об'єкт джерела даних виводу використаний для перевірки того, чи є заява фактом або виводом заяви. В програмі RDF використовує простий спрямований графік для моделювання даних, побудований шляхом ув'язування вузла джерела з вузлом призначення за допомогою названої дуги. Значення вузла джерела називається об'єктом, вузол призначення предметом, а дуга - предикатом. Ці три значення предикат, предмет і значення становлять одну заяву RDF.

Фрагмент програмної реалізації одного з SPARQL-запитів представлений нижче.

```
// Create an instance of the SPARQL-DL query engine  
engine = QueryEngine.create(manager, reasoner, true);  
// All functional ObjectProperties  
processQuery(  
"PREFIX loaf: <http://www.w3.org/TR/2003/PR-owl-guide-20031209/loaf#>\n" +  
"SELECT * WHERE {\n" +  
"ObjectProperty(?x), " +  
"Functional(?x)" +  
"}"  
);  
}  
public static void processQuery(String q)  
{  
// Create a query object from it's string representation  
Query query = Query.create(q);  
// Execute the query and generate the result set  
QueryResult result = engine.execute(query);  
}
```

При виконанні запиту на об'єкті джерела або моделі запиту вертаються в об'єкт таблиці. Клас таблиці є базовим класом для всіх об'єктів таблиці й забезпечує інтерфейс для доступу до значень у рядках і стовпцях таблиці результатів. Якщо об'єкт таблиці має

рівно три стовпці, то він може бути використаний для побудови нового об'єкта джерела даних. Це забезпечує зручний механізм для добування підмножини даних з одного джерела даних і додавання їх в іншій. У контексті RDF API вузол визначається як усі твердження про предмет URI.

Вміст таблиці порівнюємо з семантикою речення. Якщо сценарій речення не співпадає з моделлю онтології OWL, тоді існує вірогідність нетипових дій об'єкта. В такому варіанті формується відповідне повідомлення.

У додатку реалізований зручний інтерфейс з використанням SWING library, який дозволяє користувачеві виконувати всі дії у інтерактивному режимі (рис. 3). Додаток дозволяє взаємодіяти з онтологічними моделями, вже наявними в системі та завантажувати власні. Система візуалізує онтологічну модель сценарію у вигляді графа, вершинами якого являються елементи об'єкти, а ребра - зв'язки між ними. Користувач має можливість виділити окремий елемент на схем, та побачити з якими об'єктами він пов'язаний. Реалізовані й інші функції, а саме завдяки можливості власноручно створювати онтологічні вузли та семантичні триплети, додавати в існуючі онтологічні моделі.

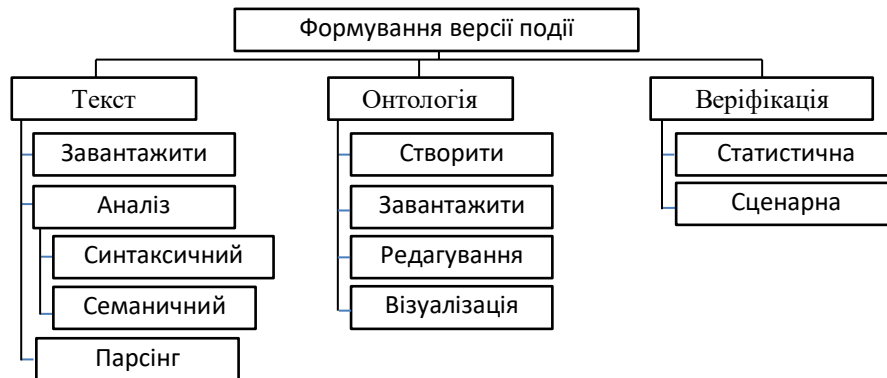


Рис. 3 – Ієрархія меню додатку

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Нижче наведено фрагмент результатів виконання аналізу тексту «Prepare a loaf. Place it under the object.» за допомогою зазначених методів класів (табл. 1).

Таблиця 1

Фрагмент результатів програмного аналізу тексту «Prepare a loaf. Place it under the object.»*

Sentence #1 (4 tokens, sentiment: Positive): Prepare a loaf. Constituency parse: (ROOT (S (VP (VB Prepare) (NP (DT a) (NN loaf))) (. .)))	Sentence #2 (6 tokens, sentiment: Neutral): Place it under the object. Constituency parse: (ROOT (S (VP (VB Place) (NP (PRP it)) (PP (IN under) (NP (DT the) (NN object)))) (. .)))
--	--

* - теги відповідають списку проекту Penn Treebank: S - просте декларативне прида-точну пропозицію, тобто таке, яке не вводиться (можливо, порожнім) подчиняючим союзом або білим словом, і яке не демонструє інверсію підлягає-дієслова, VP -глагольное

словосполучення; NP - словосполучення іменників; PP - предложное словосполучення; VB - дієслово, основна форма, DT - визначник, NN - іменник, одна або множинне, IN - привід або підрядний союз, PRP - особовий займенник

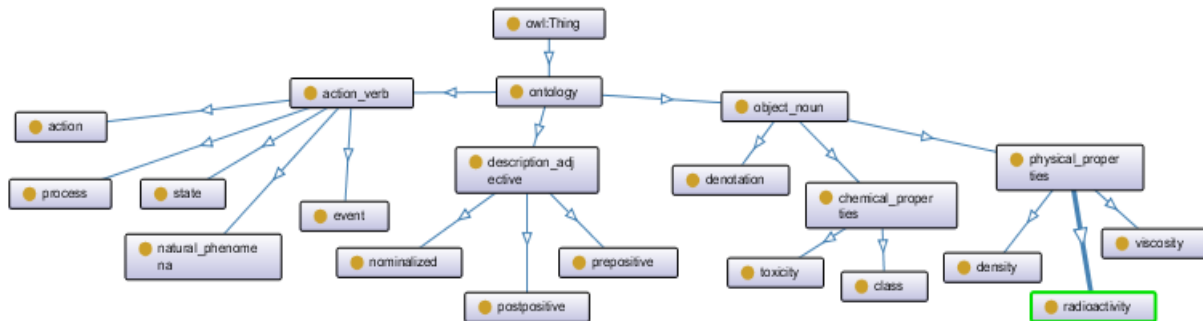


Рис. 2 – Ієрархія класу ontology

Фрагмент створеної OWL-онтології, сценарію об'єкта:

```

<rdf:RDF xml:base="">
<owl:Ontology rdf:about="urn:absolute:">
<rdfs:comment>Онтологія</rdfs:comment>
</owl:Ontology>
<owl:Class rdf:about="#action">
</owl:Class>
<owl:Class rdf:about="#action_verb">
<rdfs:subClassOf rdf:resource="#ontology"/>
</owl:Class>
<owl:Class rdf:about="#chemical_properties">
<rdfs:subClassOf rdf:resource="#object_noun"/>
</owl:Class>
.....
</rdf:RDF>
    
```

По результатах сценарної верифікації представленого тексту та моделі онтології розроблена система зробила висновок про вірогідність недопустимого сценарію та підозру про неправомірні дії.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1. Створення XML-платформи поклало початок новому наукомісткому й технологічно більш досконалому етапу в розвитку Web. Мова XML, а також інші, засновані на ньому специфікації, стали реально діючими стандартами, і багато провідних постачальників програмного забезпечення не тільки Web, але й систем баз даних, включають у свої програмні продукти підтримку мови XML. У результаті XML-платформа стає істотним компонентом у технології розробки інформаційних систем, а тенденція їх інтеграції на рівні корпорацій, відомств, міністерств тільки зміцнює позиції XML у сфері інформаційних технологій у цілому.

2. Розроблений програмний комплекс дозволяє аналізувати сценарій тексту на нетипові дії за рахунок порівняння з існуючою онтологічною моделлю. Для більш коректного використання можливостей аналізу тексту потрібно формувати корпус



онтологій або використовувати існуючі (Akutan, Amazon та т.і.) з врахуванням їх особливостей. Для збільшення онтологій об'єктів можливе додаткове використання нейромережових методів навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 *Онтология в компьютерных системах*. RSDN. <https://rsdn.org/article/philosophy/what-is-onto.xml>
- 2 Indurkha, N., & Damerau, F. J. (2010). *Handbook of Natural Language Processing*, 2nd ed. USA. Taylor and Francis Group (с. 676).
- 3 Бісікало, О. В., Висоцька, В. А. (2016). Застосування методу синтаксичного аналізу речень для визначення ключових слів україномовного тексту. *Радіоелектроніка, інформатика, управління*, (3), 54–65.
- 4 Becheru, A., Badica, C. (2014). Complex networks' analysis using an ontology-based approach: initial steps, lecture notes in artificial intelligence. *Springer international publishing Switzerland*, (8793), 326–337.
- 5 Жежерун, О. П., Яремко, О. А. (2017). Побудова рекомендаційних систем на основі онтологій. *Наукові записки НаУКМА*, (198), 36–41.
- 6 Semantic web - W3C. *World Wide Web Consortium (W3C)*. <https://www.w3.org/standards/semanticweb>
- 7 Faltin (Geneva), A. V. (2017). Natural language processing tools for computer assisted language learning. *Linguistik online*, 5/03, 137–153.
- 8 M.-C. de Marneffe et al. (2021). Universal dependencies. *Computational linguistics*, 47 (2), 255–308.
- 9 Kotov, M. (2017). NLP resources for a rare language morphological analyzer: danish case. *Proceedings of the 1st international conference computational linguistics and intelligent systems*: International Conference, Kharkiv, 31–36.
- 10 *Natural language processing tools and libraries in 2021*. Best Web & Mobile App Developers Company | The APP Solutions. https://theappsolutions.com/blog/development/nlp-tools/#contents_1
- 11 *Java API - LanguageTool Wiki*. LanguageTool Wiki (Archived) - LanguageTool Wiki. <http://wiki.languagetool.org/java-api>
- 12 *Software - the stanford natural language processing group*. The Stanford Natural Language Processing Group. <https://nlp.stanford.edu/software>
- 13 Никоненко, А. А. (2009). Обзор баз знаний онтологического типа. *Искусственный интеллект*, (4), 208–219.
- 14 Brinkmann, H., Wortarten im Deutschen. (1965). *Das Ringen um eine neue deutsche Grammatik / Darmstadt : Wissenschaftliche Buchgesellschaft*, 118–124.
- 15 *SPARQL 1.1 query language*. World Wide Web Consortium (W3C). <http://www.w3.org/TR/sparql11-query>

**Krivenko Serhii**

Doctor of Technical Sciences, Associate Professor of Systems Analysis and Information Technology
Mariupol State University, Mariupol, Ukraine
ORCID ID 0000-0002-0319-7174
s.krivenko@mdu.in.ua

Rotaniovа Natalya

Candidate of Sciences in Pedagogy, Associate Professor of Systems Analysis and Information Technology
Mariupol State University, Mariupol, Ukraine
ORCID ID 0000-0001-8437-7566
rotanevan@gmail.com

Lazarevska Yulianna

Assistant of the Systems Analysis and Information Technologies Department
Mariupol State University, Mariupol, Ukraine
ORCID ID 0000-0001-8318-5861
lazarevskayulianna@gmail.com

**AUTOMATED SYSTEM FOR DETECTION OF NON-STANDARD ACTIONS
USING SCRIPTURAL ANALYSIS OF THE TEXT**

Abstract. The scenario (narrative schemas) is some established (in society) sequence of steps to achieve the set goal and contains the most complete information about all possible ways of development of the described situation (with selection points and branches). The creation of the XML platform gave rise to a new high-tech and technologically more advanced stage in the development of the Web. As a result, the XML platform becomes a significant component in the technology of information systems development, and the tendency of their integration at the level of corporations, agencies, ministries only strengthens the position of XML in the field of information technology in general. A system for automatic detection of non-standard scripts in text messages has been developed. System programming consists of stages of ontology formation, sentence parsing and scenario comparison. The classic natural language processing (NLP) method, which supports the most common tasks such as tokenization, sentence segmentation, tagging of a part of speech, extraction of named entities, partitioning, parsing and co-referential resolution, is used for parsing sentences in the system. Maximum entropy and machine learning based on perceptrons are also possible. Ontologies are stored using OWL technology. The object-target sentence parsers with the described OWL are compared in the analysis process. From a SPARQL query on a source object, query models are returned to the table object. The table class is the base class for all table objects and provides an interface for accessing values in the rows and columns of the results table. If the table object has exactly three columns, it can be used to build a new data source object. This provides a convenient mechanism for retrieving a subset of data from one data source and adding them to another. In the context of the RDF API, a node is defined as all statements about the subject of a URI. The content of the table is compared with the semantics of the sentence. If the sentence scenario does not match the OWL ontology model, there is a possibility of atypical object actions. In this case, a conclusion is formed about the suspicion of the message. For more correct use of possibilities of the analysis of the text it is necessary to form the case of ontologies or to use existing (Akutan, Amazon, etc.) taking into account their features. To increase the ontologies of objects, it is possible to use additional neural network teaching methods.

Keywords: ontology; scenario; parser; content analysis; semantic analysis

REFERENCES (TRANSLATED AND TRANSLITERATED)

- 1 *Ontolohiya v kompiuternykh systemakh*. RSDN. <https://rsdn.org/article/philosophy/what-is-onto.xml>
- 2 Indurkha, N., & Damerou, F. J. (2010). *Handbook of Natural Language Processing*, 2nd ed. USA. Taylor



- and Francis Group (с. 676).
- 3 Bisikalo, O. V., Vysotska, V. A. (2016). Zastosuvannia metodu syntaktychnoho analizu rechen dla vyznachennia kluchovykh sliv ukrainomovnoho tekstu. *Radioelektronika, informatyka, upravlinnia*, (3), 54–65.
 - 4 Becheru, A., Badica, C. (2014). Complex networks' analysis using an ontology-based approach: initial steps, lecture notes in artificial intelligence. *Springer international publishing Switzerland*, (8793), 326–337.
 - 5 Bisikalo, O. V., Vysotska, V. A. (2016). Zastosuvannia metodu syntaktychnoho analizu rechen dla vyznachennia kluchovykh sliv ukrainomovnoho tekstu. *Radioelektronika, informatyka, upravlinnia*, (198), 36–41.
 - 6 Semantic web - W3C. *World Wide Web Consortium (W3C)*. <https://www.w3.org/standards/semanticweb>
 - 7 Faltin (Geneva), A. V. (2017). Natural language processing tools for computer assisted language learning. *Linguistik online*, 5/03, 137–153.
 - 8 M.-C. de Marneffe et al. (2021). Universal dependencies. *Computational linguistics*, 47 (2), 255–308.
 - 9 Kotov, M. (2017). NLP resources for a rare language morphological analyzer: danish case. *Proceedings of the 1st international conference computational linguistics and intelligent systems*: International Conference, Kharkiv, 31–36.
 - 10 *Natural language processing tools and libraries in 2021*. Best Web & Mobile App Developers Company | The APP Solutions. https://theappsolutions.com/blog/development/nlp-tools/#contents_1
 - 11 *Java API - LanguageTool Wiki*. LanguageTool Wiki (Archived) - LanguageTool Wiki. <http://wiki.languagetool.org/java-api>
 - 12 *Software - the stanford natural language processing group*. The Stanford Natural Language Processing Group. <https://nlp.stanford.edu/software>
 - 13 Nykonenko, A. A. (2009). Obzor baz znanyi ontolohycheskoho typu. *Yskusstvennyi yntellekt*, (4), 208–219.
 - 14 Brinkmann, H., Wortarten im Deutschen. (1965). *Das Ringen um eine neue deutsche Grammatik / Darmstadt : Wissenschaftliche Buchgesellschaft*, 118–124.
 - 15 *SPARQL 1.1 query language*. World Wide Web Consortium (W3C). <http://www.w3.org/TR/sparql11-query>

