

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНА РОБОТА

Метод та програмні засоби препроцесингу вхідного текстового контенту
Назва теми

Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр ДРПЗ. 2001113.20.02.06ПЗ

Виконала студентка
2 курсу групи ПЗм-20-2 _____

Керівник канд. техн. наук, доцент
Науковий ступінь, звання

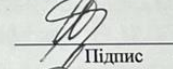
Нормоконтролер канд. техн. наук, доцент

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення

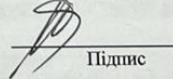
7 грудня 2021 р.


Підпис

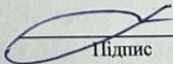
О.Г.Онишко
Ініціали, прізвище


Підпис

О. М. Яшина
Ініціали, прізвище


Підпис

О. М. Яшина
Ініціали, прізвище


Підпис

Л. П. Бедратюк
Ініціали, прізвище

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри Л. П. Бедратюк
02 09 2021 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)**

Онишко Оксані Григорівні

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Методи та програмні засоби Препроцесінгу вхідного текстового контенту

Керівник проєкту (роботи) Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2021 р. № 118

2. Строк подання студентом проєкту (роботи) на кафедру 01.12.2021 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1 Дослідження предметної області

2 Моделі та методи для вирішення проблеми





3 Проектування програмного забезпечення для вирішення проблеми

4 Реалізація програмного забезпечення для вирішення проблеми

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|--|--|
| | | завдання видав | завдання прийняв |
| Антиплагіат | Гурман І.В., доцент |  23.11.21 |  01.12.21 |
| Нормоконтроль | Яшина О.М., доцент |  23.11.21 |  01.12.21 |

7. Дата видачі завдання « 01 » вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

| Назва етапів (розділів) дипломного проєкту (роботи) | Строк виконання етапів проєкту (роботи) | Примітка |
|---|---|----------|
| 1. Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження; визначення структури дипломної роботи | 01.09 - 07.09.2021 | |
| 2. Визначення структурних і функціональних особливостей предметної області, аналіз відомих рішень поставленої проблеми, постановка задачі. | 08.09 - 25.09.2021 | |
| 3. Розробка методів та моделей для вирішення поставленої задачі | 26.09 - 10.10.2021 | |
| 4. Робота над науковими статтями | 11.10 - 20.10.2021 | |
| 5. Проектування архітектури програмної системи, опис вимог до її функціонування, проектування бази даних, інтерфейсу та вибір технологій для реалізації | 11.10 - 26.10.2021 | |
| 6. Детальний опис програмної реалізації системи та опис результатів тестування | 27.10 - 15.11.2021 | |
| 7. Узгодження постановки задачі, отриманих результатів та висновків; написання вступу, загальних висновків, оформлення джерел посилання та додатків; оформлення пояснювальної записки згідно вимог стандартів | 16.11 - 30.11.2021 | |
| 8. Попередній захист дипломної роботи | Листопад (згідно графіка) | |
| 9. Перевірка роботи на наявність плагіату; норм контроль; брошурування пояснювальної записки; підготовка супровідних документів | 01.12 - 04.12.2021 | |
| 10. Підготовка до захисту дипломної роботи | 06.12 - 08.12.2021 | |

Студент

Підпис

О.Г.Онишко

Ініціали, прізвище

Керівник проєкту (роботи)

Підпис

О. М. Яшина

Ініціали, прізвище

РЕФЕРАТ

Тема роботи: Методи та програмні засоби препроцесингу вхідного текстового контенту.

Автор роботи: Онишко Оксана Григорівна

Керівник роботи: Яшина Оксана Миколаївна

Обсяг – 82 с., 13 рис., 2 табл., 7 додатків, 19 джерел.

ПРЕПРОЦЕСИГ, СИНТАКСИЧНИЙ МЕТОД, СЕМАНТИКО-СИНТАКСИЧНИЙ МЕТОД, ПРЕДИКАТ, ПРИРОДНЯ МОВА

Об'єктом дослідження є процес створення семантико-синтаксичних методів аналізу текстів українською мовою..

Предметом дослідження є методи і алгоритми визначення рольових структур тверджень, а також методи і алгоритми семантичного розбору.

Метою роботи є удосконалення якості методів та програмних засобів препроцесингу вхідного текстового коду за допомогою синтаксичного та семантичного аналізу текстів природною мовою шляхом розробки методу семантико-синтаксичного аналізу.

Наукова новизна отриманих результатів:

1. Удосконалено метод автоматичного визначення рольових структур висловлень, заснованих на комунікативній граматиці української мови.
2. Удосконалено метод комп'ютерного семантико-синтаксичного аналізу текстів, в який інтегровані методи побудови синтаксичних дерев залежності і визначення рольових структур висловлень. Застосування розробленого методу надає можливість покращити точність та повноту синтаксичного та семантичного аналізу.

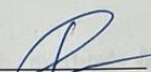
На основі розроблених методів було створено програмний засіб препроцесингу вхідного текстового коду.

Дослідження в першу чергу спрямоване на створення методів аналізу текстів українською мовою.

Результати дослідження показують, що розроблено новий метод автоматичного визначення рольових структур висловлень, заснований на комунікативній граматиці української мови.

Практичне значення отриманих результатів полягає в тому, що експериментально показано, що використання рольової структури речення підвищує ефективність побудови правил вилучення визначень та термінів автора з текстів наукових публікацій.

3.12.21
(дата)


(підпис)

ABSTRACT

Theme: Methods and software for preprocessing incoming text content.

Author of the work: Onyshko Oksana Grigorievna

Head of work: Oksana Yashyna

Volume - 82 p., 13 figures, 2 table., 7 annexes, 19 sources.

PREPROCESSING, SYNTACTIC METHOD, SEMANTIC-SYNTACTIC METHOD, PREDICATE, NATURAL LANGUAGE

The object of research is the creation of semantic-syntactic methods for analyzing texts in Ukrainian. .

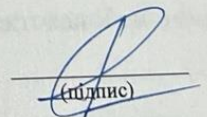
The subject of the study are methods and algorithms for determining the role structures of statements, as well as methods and algorithms of semantic analysis.

The purpose of the work is to improve the quality of methods and software for preprocessing input text code using syntactic and semantic analysis of texts in natural language by developing a method of semantic and syntactic analysis in which these two types of analysis are integrated.

The results of the study show that a new method of automatic determination of role structures of expressions based on the communicative grammar of the Ukrainian language has been developed.

The practical significance of the results is that it is experimentally shown that the use of the role structure of a sentence increases the effectiveness of building rules for removing the definitions and terms of the author from the texts of scientific publications.

3.12.21
(дата)


(підпис)

ЗМІСТ

| | |
|---|----|
| Вступ..... | 7 |
| 1 Дослідження предметної області..... | 7 |
| 1.1 Характеристика структурних і функціональних особливостей предметної області..... | 7 |
| 1.2 Аналіз відомих моделей, методів та засобів..... | 8 |
| 1.3 Висновки. Постановка задачі..... | 20 |
| 2 Моделі та методи для вирішення проблеми..... | 22 |
| 2.1 Інтеграція методів синтаксичного і семантичного аналізу текстів..... | 22 |
| 2.2 Метод визначення рольових структур тверджень у текстах українською мовою..... | 24 |
| 2.3 Метод семантико-синтаксичного аналізу текстів..... | 34 |
| 2.4 Висновки..... | 44 |
| 3 Алгоритми та технології вирішення задачі..... | 46 |
| 3.1 Використання семантичного і семантико-синтаксичного аналізу при побудові семантичної мережі реляційно-ситуаційної моделі..... | 46 |
| 3.2 Застосування семантичного і семантико-синтаксичного аналізу при запитально-відповідальному пошуці..... | 48 |
| 3.3 Застосування семантичного та семантико-синтаксичного аналізу для вирішення проблеми автоматичного вилучення визначень та термінів автора з текстів наукових публікацій..... | 53 |
| 3.4 Аналіз і вибір технологій та методів реалізації проектованої системи..... | 56 |
| 3.5 Висновки..... | 57 |
| 4 Реалізація методів семантичного та семантико-синтаксичного аналізу..... | 59 |
| 4.1 Використання семантичного і семантико-синтаксичного аналізу при побудові семантичної мережі реляційно-ситуаційної моделі..... | 59 |
| 4.2 Експериментальне дослідження методу пошуку семантичних аргументів... .. | 60 |
| 4.3 Результати експериментів методу семантичного аналізу..... | 62 |
| 4.4. Результати експериментів методу семантико-синтаксичного аналізу..... | 66 |

| | |
|--|-----|
| 4.5 Оцінка ефективності методу семантико-синтаксичного аналізу на якість вирішення задачі визначення рольових структур тверджень | 71 |
| 4.6 Програмна реалізація | 73 |
| 4.7 Висновки | 73 |
| Висновки | 75 |
| Перелік джерел посилання | 76 |
| Додаток А Програмний код основних модулів..... | 80 |
| Додаток В Копії наукових публікацій..... | 92 |
| Додаток С Презентаційні матеріали..... | 116 |

ВСТУП

Актуальність дослідження. Успішне розв'язання задач ідентифікації, прогнозування та діагностики неможливе без попередньої формалізації задачі та підготовки даних тобто їх препроцесингу. Тому залучаються різні види аналізу природної мови. Однією із прогресивних галузей яка розглядає ці задачі є комп'ютерний лінгвістичний аналіз.

Комп'ютерний лінгвістичний аналіз текстів природної мови є перспективною галуззю штучного інтелекту, яка швидко розвивається. Одним з ключових завдань обчислювального лінгвістичного аналізу є створення такого структурованого представлення тексту, до якого можуть бути застосовані методи і алгоритми рішення прикладних задач. Багато дослідників звертають увагу на те, що якісне вирішення таких задач, як пошук за допомогою питань, вилучення інформації і знань з текстів, автоматичне абстрагування, неможливе без проведення синтаксичного і семантичного аналізу.

Існує велика кількість різновидів методів як синтаксичного, так і семантичного аналізу, які спираються на різні моделі синтаксичної структури речення із різним розумінням семантики.

Дерева залежності створюють синтаксичну структуру речень як ієрархію слів, з'єднаних дугами, які позначають синтаксичне підпорядкування між головним і залежним словами. Підпорядкування обумовлено набором загальних принципів, які визначають, що залежне слово в реченні є уточнююче, необов'язкове, менш важливе для передачі значення твердження, ніж головне.

Семантика моделі, для якої за основу взята рольова структура речення, що дозволяє абстрагуватися від синтаксичних дерев речень і порівнювати одні і ті ж семантичні структури з різними граматичними конструкціями. Завдання визначення рольових структур тверджень обирає пошук предикатних слів, які описують певну ситуацію в реченні, пошук їх семантичних аргументів -

синтаксичних конструкцій, які показують у реченні учасників ситуації, а також визначення значень аргументів, тобто присвоєння їм семантичних ролей, які відповідають учасникам в даній ситуації.

Проблеми синтаксичного і семантичного аналізу, як правило, не вирішуються разом: спочатку будується синтаксичну структуру речення, на основі якої потім будується семантична структура. Для цього існує ряд методів, що використовують як правила, так і машинне навчання. Хоча сучасні методи дозволяють домогтися досить якісного вирішення цих проблем, але завжди залишається значний простір для вдосконалення цих методів. Аналіз помилок у синтаксичних і семантичних аналізаторах показує, що, з одного боку, для побудови правильної синтаксичної структури речення необхідно знання його семантики, а з іншого боку, помилки в дереві синтаксису можуть негативно впливати на якість семантичного аналізу. Виходячи з цього, деякі дослідники мають припущення, що поєднання синтаксичного і семантичного типів аналізу може поліпшити їх якість. Такий комбінований підхід до вирішення проблеми глибокого лінгвістичного аналізу назовемо семантично-синтаксичним аналізом.

Об'єктом дослідження є процес створення семантико-синтаксичних методів аналізу текстів українською мовою..

Предметом дослідження є методи і алгоритми визначення рольових структур тверджень, а також методи і алгоритми семантичного розбору.

Метою роботи є удосконалення якості методів та програмних засобів препроцесингу вхідного текстового коду за допомогою синтаксичного та семантичного аналізу текстів природною мовою шляхом розробки методу семантико- синтаксичного аналізу.

Наукова новизна отриманих результатів:

1. Удосконалено метод автоматичного визначення рольових структур висловлень, заснованих на комунікативній граматиці української мови.
2. Удосконалено метод комп'ютерного семантико-синтаксичного аналізу текстів, в який інтегровані методи побудови синтаксичних дерев залежностей і визначення рольових структур висловлень. Застосування

розробленого методу надає можливість покращити точність та повноту синтаксичного та семантичного аналізу.

На основі розроблених методів було створено програмний засіб препроцесингу вхідного текстового коду.

Дослідження в першу чергу спрямоване на створення методів аналізу текстів українською мовою.

Результати дослідження показують, що розроблено новий метод автоматичного визначення рольових структур висловлень, заснований на комунікативній граматиці української мови.

Практичне значення отриманих результатів полягає в тому, що експериментально показано, що використання рольової структури речення підвищує ефективність побудови правил вилучення визначень та термінів автора з текстів наукових публікацій.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика структурних і функціональних особливостей предметної області

Синтаксичний аналіз займає одне з провідних місць в ланцюжку обробки текстів природною мовою у багатьох програмних додатках. Він використовується в інформаційно-пошукових та інформаційно-аналітичних системах, системах машинного перекладу та для вилучення інформації з текстів. На сьогоднішній день синтаксичний аналіз є актуальною і активно досліджуваною проблемою комп'ютерної лінгвістики.

У сфері обробки текстів на природній мові, під синтаксичним аналізом називається порівняння лексем певної мови (природної або формальної) з її правописом. Одиницею аналізу текстів на природній мові як правило є речення. З одного боку, синтаксичний аналіз є вирішенням проблеми розпізнавання. Він визначає, чи є речення граматично правильним з точки зору загальноприйнятих правил побудови висловлень на будь якій природній мові. Наприклад, в англійській мові артикль ставиться перед словом, в українській мові головне і залежне слово повинні бути однакові за родом, числом і відмінком. Такі вимоги є в системах автоматичної перевірки якості тексту в текстових редакторах. З іншого боку, що більш важливо, в задачі розпізнавання тексту машиною синтаксичний аналіз – це побудова такої структури, яка б дозволяла наблизитися до деякого явного формалізованого представлення значення тексту. Але, на відміну від «глибокої» семантичної структури, яка будується в результаті семантичного аналізу, синтаксична структура зазвичай не пов'язує конструкцію природньої мови з їх значеннями в певній предметній області. Синтаксична структура може бути або як проміжний результат, що є входом для семантичного аналізу, або як зручне представлення тексту на природній мові для вирішення високорівневих прикладних задач. Наприклад, в інформаційно-аналітичних системах або системах машинного перекладу [8, 9]. Більшість створених моделей синтаксичних структур

речення природної мови спираються або на граматику компонентів, запропонованих у роботах Ноама Хомського [10], або на граматику залежностей, для яких фундаментальними вважаються роботи Люсьєна Теньєра [11] та Ігоря Мельчука [12].

2.2 Аналіз відомих моделей, методів і засобів

На початковому етапі досліджень в області комп'ютерної лінгвістики основна увага приділялася граматиці складових. Ця модель передбачає, що речення природної мови можуть бути представлені як ієрархія складових – проєктивних синтаксичних груп, які не можуть частково перетинатися, але які, в свою чергу, складаються з менших груп, аж до атомарних груп – слів речення. Таку ієрархічну структуру називається деревом складових (рисунок 1.1).

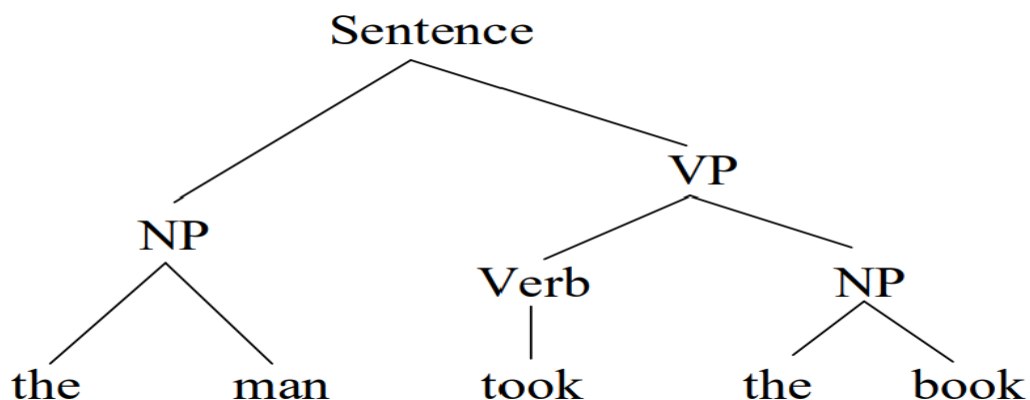


Рисунок 1.1 – Перше дерево складових, з роботи Н. Хомського [10]

Ідея про те, що слова в реченнях природної мови згруповані в складові, заснована на лінгвістичному спостереженні, що ланцюжки слів у реченні можуть функціонувати як єдине ціле і підпорядковуються єдиним граматичним правилам. Складові можуть бути перенесені в середину або кінець речення в цілому, але частково без втрати змісту – ні.

Граматики складових в ієрархії Хомського це контекстно вільна граматики. Дослідники, в основному, сходяться на думці, що природні мови не є регулярними, наприклад, в роботах [10, 13, 14] показано, що англійська мова не може бути розпізнана регулярними граматами. Дослідники також дійшли висновку, що існують мови, які містять конструкції, що можуть бути розпізнані тільки за допомогою контекстно-залежних грамастик. Наприклад, у швейцарській німецькій мові є конструкції, що не розпізнаються контекстно-вільними граматами. Однак, якщо природні мови можуть виходити за рамки класу контекстно-вільних мов, моделювання таких синтаксичних правил з граматики за допомогою контекстно-вільних грамастик дає досить непогане наближення до реальності і дозволяє вирішити більшість прикладних задач.

Граматики залежності передбачає, що речення тексту можуть бути структуровані як дерева залежностей, де слова пов'язані орієнтованими дугами, які позначають синтаксичне підпорядкування між головним і залежним словом [17, 12, 18] (Рисунок 1.2).

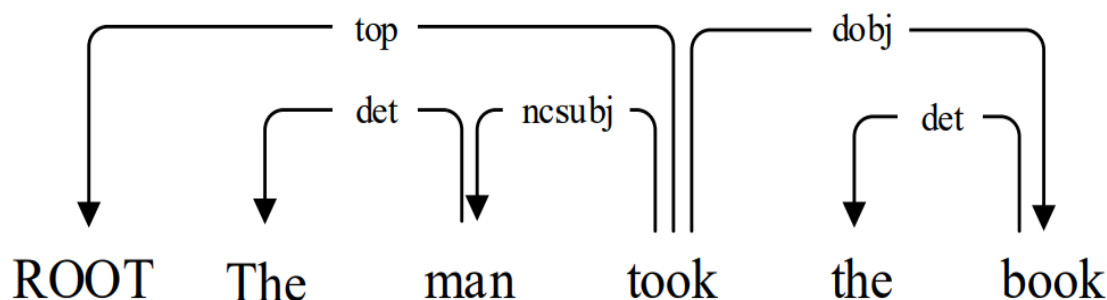


Рисунок 1.2 – Синтаксичне дерево залежностей речення «The man took the book» у форматі ресурсу «Universal Dependencies» [19]

Головна відмінність синтаксичних дерев залежностей від дерев складових полягає в тому, що в них відсутні нетермінальні вершини, які позначають складові, а синтаксичні зв'язки мають помітки, які позначають їх тип. Типи зв'язків визначають граматичні функції слів у реченні або загальні семантичні відношення між словами.

Наявні граматичні теорії залежностей можуть відрізнитися в тому, за якими правилами встановлюють зв'язки, які слова в цих зв'язках головні, а які залежні, та які є типи синтаксичних зв'язків. Можна навести досить багато традиційно суперечливих випадків але тим не менше існує ряд загальних принципів, які дають пояснення наявності синтаксичної залежності між словами та виразами на природній мові. У [20] Йоахім Нівре наводиться деякі з них:

- головне слово визначає синтаксичну категорію виразу і часто може замінити його;
- головне слово визначає семантичну категорію виразу, а залежне визначає семантичну характеристику;
- головне слово – обов'язкове, залежне – може бути необов'язковим;
- головне слово виділяє залежне слово і визначає, чи потрібно воно;
- форма залежного слова узгоджується з формою головного;
- позиція залежного слова залежить від головного.

Вибір дерев залежностей або дерев складових для опису синтаксичної структури речень природньої мови пов'язаний з двома властивостями мов: проєктивністю їх синтаксичних конструкцій і наявністю вільного порядку слів.

Властивість проєктивності в термінах синтаксичного дерева залежностей означає: якщо всі стрілки залежностей проведені з одного боку від прямої, на якій записано речення та жодна з стрілок не перетинає жодної іншої стрілки, і ніяка стрілка не накриває кореневий вузол [21, 22]. В термінах дерев, що складають проєктивність означає, що складові не можуть розриватись на декілька окремих, несуміжних частин.

Граматики складових широко використовуються для аналізу мов з фіксованим порядком слів, в яких мало конструкцій, які порушують властивість проєктивності, оскільки в граматиці складових, як правило, не передбачені непроєктивні синтаксичні відношення між словами. Такими мовами є англійська, німецька, турецька.

Вважається, що граматики залежностей добре відображають специфіку мов з вільним порядком слів, в яких між словами може бути присутня значна кількість

непроективних зв'язків. До таких мов належать німецька, чеська, російська та інші східнослов'янські мови.

Відзначимо, що наявність вільного порядку слів в природній мові також впливає на трудомісткість його опису з використанням різних формалізмів. При аналізі конструкцій мов з вільним порядком слів в граматиках складових може вимагатися велика кількість правил окремо для кожного можливого випадку розміщення слів [23], тоді як правила граматики залежностей часто абстрагуються від порядку слів

Методи побудови синтаксичних дерев складових.

Як видно із вищезазначеного, якщо природні мови з достатнім ступенем узагальнення можна наблизити до контекстно-вільних мов то, для їх синтаксичного аналізу можуть використовуватись контекстно-вільні граматики і відповідні алгоритми для їх розпізнавання. Але від формальних контекстно-вільних мов, таких як мови програмування, які зазвичай можуть бути ефективно проаналізовані в лінійний час від довжини вхідного рядка, природні мови характеризуються високою двозначністю [23].

Двозначність може виникнути на всіх рівнях обробки текстів в природній мові. На етапі морфологічного аналізу може виникнути декілька варіантів розбору слів, які мають різні морфологічні характеристики.

Контекстно-вільна граMATика природньої мови як правило містить велику кількість недетермінованих правил, що при розборі за допомогою класичних алгоритмів призводить до великої кількості повернень і, як наслідок, до надзвичайно низької обчислювальної ефективності аналізу. Крім того, граMATика природньої мови зазвичай допускає декілька можливих варіантів розбору, тобто. допускає синтаксичну (або структурну) омонімію.

Часто цей тип омонімії не може бути вирішений тільки лінгвістичними міркуваннями і вимагає знань семантики слів для визначення відповідного розбору і конструкцій. Відзначимо, що бувають випадки, коли декілька варіантів синтаксичного розбору правильні з точки зору семантики. Наприклад, у реченні

«Ми зустрічали лікаря з Німеччини» обидва аналізи, представлені на рисунку 1.3, допустимі з точки зору семантики.

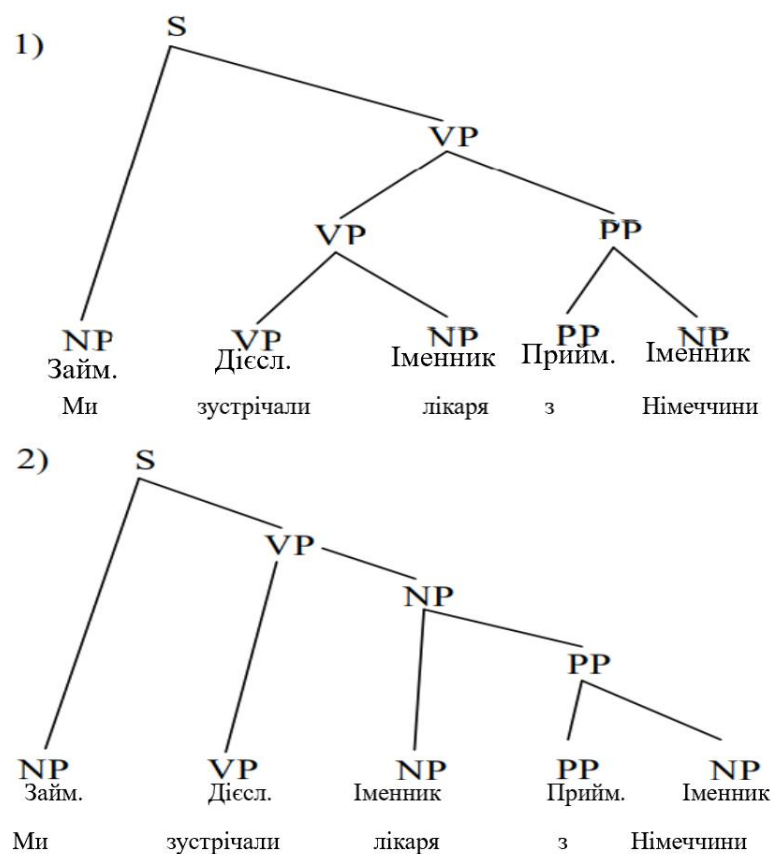


Рисунок 1.3 – Приклад двох варіантів розбору речення «Ми зустрічали лікаря з Німеччини», допустимого з точки зору семантики

Для вирішення омонімії в таких випадках потрібні знання про прагматику тексту або про його екстралінгвістичний контекст [24]. При відсутності знань семантики і прагматики тексту аналізатор повинен або видати всі можливі варіанти розбору, або прагнути знайти оптимальний варіант відповідно до якогось критерію, заснованого на вже побудованих морфологічних або синтаксичних структурах речень. Таким чином, двозначність природної мови робить класичні алгоритми розбору контекстно-вільної граматики непридатними.

Щоб впоратися з двома проблемами, які виникають при аналізі текстів на природній мові з використанням контекстно-вільної граматики, а саме з великою кількістю повернень і необхідністю збереження декількох варіантів аналізу,

дослідники запропонували ряд алгоритмів з використанням принципів динамічного програмування. Основна ідея використання динамічного програмування в аналізі за допомогою контекстно-вільної граматики полягає в тому, щоб компактно зберігати всі піддерева, побудовані в ході аналізу, так щоб при поверненні не було необхідності знову створювати вже побудовані піддерева.

Одним з найпростіших алгоритмів синтаксичного аналізу, які використовують принципи динамічного програмування, є алгоритм Cocke - Younger – Kasami. Для його роботи необхідно перетворити граматику в нормальну форму Хомського, в якій всі правила з правого боку мають або термінал, або пару нетерміналів. Ця форма дозволяє при розборі використовувати верхньотрикутну таблицю для зберігання всіх розпізнаних нетерміналів (синтаксичних груп). Кожна комірка $[i, j]$ позначає складову, яка включає в себе токени з i -тої позиції по j -ту, де зберігаються всі розпізнані нетермінали цієї складової. Розпізнавання нетерміналів в комірці $[i, j]$ здійснюється шляхом співставлення нетерміналів з комірок лівіше та комірок нижче вказаної з правилами граматики. Алгоритм проводить аналіз знизу вгору, спочатку заповнюючи ліву і нижні комірки. Речення вважається розпізнаним, якщо у верхній правій клітинці введено нетермінал S , що позначає речення. Недоліком оригінального алгоритму СУК є те, що він працює тільки з контекстно вільними граmaticами зведеними до нормальної форми Хомського. Тому для отримання дерева складових з термінами початкової граматики необхідно обернене перетворення, що вимагає додаткових обчислювальних затрат і затрат пам'яті.

Інший алгоритм з використанням принципів динамічного програмування був запропонований Джеєм Ерлі [27]. Цей алгоритм виконує розбір зверху вниз. Алгоритм використовує таблицю стану для зберігання проміжного дерева. Під станом розуміється «правило з крапкою», тобто правило, в правій частині якого відмічено, наскільки воно виконано в заданому положенні в тексті. Алгоритм послідовно переглядає речення і заповнює таблицю стану зліва направо. Розбір починається з додавання в ліву клітинку таблиці правила, в лівій частині якої знаходиться нетермінал S . Потім алгоритм, в залежності від поточних станів і

поточного слова, виконує дії: прогнозування, сканування, виконання. Розбір вважається виконаним, якщо в останньому записі таблиці є правило, в лівій частині якого є нетермінал S.

Мартін Кей об'єднав два вищенаведених підходи, представивши схему побудови ефективних алгоритмів розбору контекстно-вільної граматики – «chart parser» [28]. У цій схемі він запропонував зберігати частково побудовані дерева у вигляді графа, а також ввів поняття стратегії. Змінюючи стратегію, можна побудувати алгоритм розбору знизу-вгору типу СҮК або зверху-вниз типу Ерлі, чи реалізувати більш складну стратегію, яка, наприклад, на основі ряду евристик могла б вибрати наступну дію, змінивши напрямок розбору.

Алгоритми СҮК, Ерлі та інші різновиди «chart parser» в гіршому випадку мають складність $O(n^3)$, де n - кількість слів у реченні.

Створення контекстно-вільних граматик для аналізу природніх мов є дуже трудомістким завданням. Для використання в реальних додатках граматики повинні містити тисячі правил. Підтримувати і налагоджувати ці граматики дуже складно. Розглянемо статистичні підходи, які широко використовуються для ранжування варіантів розбору

Найвідомішим статистичним підходом до проблеми побудови компонентних дерев є використання стохастичних контекстно вільних граматик [30]. Стохастичні контекстно-вільні граматики відрізняються від звичайних контекстно-вільних граматик тим, що кожному граматичному правилу присвоюється ймовірність. Задачею синтаксичного аналізатора в цьому випадку є знаходження найбільш правильного варіанту розбору речення. Стохастичні контекстно-вільні граматики автоматично будуються на основі сховищ синтаксичних дерев – корпусів, в яких реченням в ручну або напівавтоматично зіставлені синтаксичні дерева. Наприклад, для англійської мови найвідомішим таким банком є Penn Treebank [31]. Для розбору речень на природній мові по стохастично контекстним вільним граматакам в основному використовують модифікації алгоритмів, раніше представлених з використанням принципів динамічного програмування (СҮК, Ерлі, «chart parser»). Запропоновані дослідниками підходи відрізняються в основному методами

моделювання ймовірностей синтаксичних дерев, а також способами вирішення проблеми знаходження дерева з максимальною ймовірністю.

Перші стохастичні контекстно-вільні граматики в основному спиралися на синтаксичні правила взаємодії частин мови. Однак експерименти показали, що статистичної інформації про сумісність частин мови недостатньо для вирішення неоднозначності природної мови. Оцінки якості таких аналізаторів були значно нижчими, ніж у аналізаторів з використанням великих наборів евристик. Розвитком статистичного підходу стали лексиколізовані стохастичні контекстно-вільні граматики [32]. У правилах цих граматики, крім частин мови, враховують також словниковий запас і статистику слів, що зустрічаються. Ця модифікація значно покращила якість методів статистичного аналізу.

Семантичний аналіз. Комп'ютерний семантичний аналіз тісно пов'язаний з задачею розуміння тексту машиною. Існує багато інтерпретацій поняття «зміст тексту» і задачі його розуміння. Наприклад по Д.А. Поспелову система розуміє введений в неї текст, якщо з точки зору людини вона правильно відповідає на питання, які пов'язані з інформацією, яка знаходиться в тексті.

Робота Р. Монтегю породила цілий напрямок, в рамках якого досліджуються підходи до моделювання формальною мовою часових аспектів природної мови, виразів подій, характеристик об'єктів в тексті, ідіом, метафор і т. д. На практиці відомі спроби застосування формальної семантики в задачі машинного перекладу [72]. Однак, на сьогоднішній день дослідники прийшли до висновку, що через двозначність і гнучкість природної мови використання апарату формальної семантики на практиці, з одного боку, є проблематичним через його складність, а з іншого боку, недостатньою для вирішення сучасних проблем обробки текстів на природній мові.

Семантико-синтаксичний аналіз. Ідея інтеграції комп'ютерного синтаксичного і семантичного аналізу була висловлена багатьма дослідниками [25,39]. На сьогоднішній день цілком очевидно, що через неоднозначність конструкцій в природній мові для побудови синтаксичних дерев з високою точністю неможливо обмежитися тільки знанням граматики мови у вузькому сенсі

– тобто лінгвістичними знаннями про морфологію слів і про синтаксичні правила аналізу структур, які діють тільки з такими лінгвістичними категоріями, як частини мови і морфологічні признаки. Для того, щоб вирішити цю проблему, в загальному випадку потрібні статистичні знання про природну мову, знання семантики природної мови.

Структурна неоднозначність природної мови вимагала розробки спеціальних аналізаторів, статистичних методів і підходів з використанням методів машинного навчання для синтаксичного аналізу. У багатьох роботах з розбору відзначається велике значення лексичних ознак для досягнення високих показників якості аналізу. Тому сучасні синтаксичні аналізатори текстів на природній мові часто враховують не тільки поєднання частин мови, морфологічних особливостей і структурних особливостей, але і лексичну сумісність слів один з одним, що в певному сенсі є формою пізнання семантики мови.

Але є підходи, які йдуть набагато далі, ніж просто облік лексем, лем або семантичних класів слів на етапі синтаксичного аналізу. Деякі дослідники пропонують альтернативи традиційному підходу, де розбір, в якому побудовано дерево синтаксису, строго передуює семантичному аналізу, в якому формується семантичне представлення тексту, і розробляє підходи, які об'єднують ці два типи аналізу в одній процедурі. Аналіз тексту на природній мові, який виконується відповідно до цих підходів, має назву семантико-синтаксичним.

Семантико-синтаксичний аналіз поєднує в собі підходи, які використовують як тісну інтеграцію синтаксичного, так і семантичного аналізу, наприклад, в єдиному алгоритмі, так і підходи, в яких вони розділені, але результати, отримані на двох етапах аналізу і впливають один на одного.

Прагнення до тісної інтеграції ґрунтується на припущенні, що наявність доступу під час аналізу до синтаксичних та семантичних знань про природну мову, а також до обох частково побудованих структур дозволить вирішити мовленнєві неоднозначності раніше, ніж у традиційному послідовному підході. З одного боку, це може призвести до переваги в обчислювальній ефективності обробки, оскільки в спільному аналізі не потрібно генерувати непотрібні варіанти розбору та

зберігати їх до тих пір, поки двозначність не буде вирішена в семантичному аналізі. Але, з іншого боку, паралельний семантичний аналіз може значно уповільнити загальний аналіз. Так як багато неправильних варіантів розбору можна усунути тільки шляхом синтаксичного аналізу, для таких варіантів виконання паралельного семантичного аналізу перетворюється в додаткові накладні витрати.

Модель семантики тексту на основі рольової структури речення. Однією з найдавніших моделей семантики тексту на природній мові є модель, заснована на рольовій структурі речення. Сучасне формулювання цієї моделі було запропоновано в роботах Чарльза Філмора. Він ввів поняття глибинного відмінку – універсальної (тематичної) ролі, яку можуть виконувати учасники ситуації, заявленої в тексті дієсловом. Глибинні відмінки запропоновано розглядати як «універсальні вроджені поняття, які ідентифікують типи суджень, які людина здатна виносити про події, що відбуваються навколо нього, – судження про такі речі, як «хто щось зробив», «з ким щось сталося», «що зазнало якихось змін». Всього було введено близько дванадцяти таких понять, які, як і очікувалося, дозволяють представити в узагальненому вигляді всі можливі розуміння змісту подій учасниками ситуацій. На сьогоднішній день немає згоди між дослідниками про склад засобів для ролей і про те, на яких принципах їх слід виділяти і розрізняти.

Модель семантики тексту, заснована на рольовій структурі речення, була розроблена в багатьох роботах як вітчизняних, так і зарубіжних лінгвістів. Речення тексту в цій моделі можуть бути представлені у вигляді набору предикатних слів, їх аргументів і семантичних ролей [22].

1. Предикатне слово відноситься до лексеми або синтаксичної конструкції, що має набір семантичних валентностей. Предикатне слово в тексті позначає ситуацію.

2. Семантичні аргументи – це мовні вирази в тексті, якими заповнюються семантичні валентності предикатних слів.

3. Семантична роль є частиною семантики предикату, що відображає загальні властивості аргументу предиката.

4. Предикатне слово та його аргументи формують предикатно аргументну структуру. Між предикатним словом і його семантичним аргументом встановлено предикатно-аргументний зв'язок.

5. Однією з найважливіших властивостей моделі семантики тексту, заснованої на рольовій структурі речення, є те, що зазвичай (за винятком випадків з симетричними дієсловами) одне предикатне слово не може мати два аргументи з однаковою семантичною роллю, а аргументи можуть мати лише одну роль з заданим предикатним словом.

Модель семантики тексту, заснована на рольовій структурі речення, розвивалася як в напрямку узагальнених ролей [80], так і в напрямку більш специфічних ролей, визначених для конкретних предикатних слів або груп предикатних слів – фреймів. У сучасних системах обробки текстів для багатьох мов найчастіше використовуються моделі семантики ресурсів PropBank і FrameNet.

Модель ресурсу PropBank передбачає, що кожне значення предикату має набір специфічних для нього ролей. Ролі в PropBank не мають назви, а тільки пронумеровані: Arg0, Arg1, ..., Arg5. Ролям Arg0 і Arg1 зазвичай відповідають узагальнення агенса і пацієнта, значення інших ролей залежить від предикатних слів. На додаток до пронумерованих ролей, PropBank надає узагальнені ролі Argm-*, які є універсальними для всіх предикатних слів. Наприклад, Argm-loc відповідає місцю події; Argm-dir – напрямку дії; Argm-tnp – часу або тривалості дії. Ресурс PropBank складається з текстів банку синтаксичних дерев Penn Treebank, позначених семантичними ролями дієслів, і словника, який перераховує пронумеровані ролі з коментарями та прикладами для англійських дієслів. Ресурс NomBank забезпечує аналогічну розмітку для предикатів слів іменників. Є багато ресурсів, подібних до оригінального PropBank іншими мовами (наприклад, китайська, арабська).

На відміну від PropBank, FrameNet визначає семантичні ролі не для окремих дієслів, а для фреймів, які об'єднують в собі кілька предикатів слів, близьких за значенням. Слова різних частин мови можуть потрапляти в один фрейм: дієслова, іменники, прислівники і т. д. Ролі в FrameNet діляться на центральні - специфічні для конкретного фрейму і периферійні – можуть виникати в різних ситуаціях (час,

місце, тривалість і т. д.). Ще однією важливою особливістю FrameNet є те, що він фіксує зв'язки між різними кадрами та семантичними ролями. Фрейми можуть бути пов'язані відношеннями слідування, причинно-слідування, перспективними зв'язками (один фрейм розглядає подію з вужчої точки зору, ніж інший нейтральний фрейм), частина-ціле і т.д. FrameNet, як і PropBank, надає семантично позначений корпус текстів англійською мовою. Існують ресурси типу FrameNet для інших мов, наприклад, для німецької, для іспанської.

Реляційно-ситуативна модель. Реляційно-ситуативні моделі тексту формалізують глибокі семантичні структури тексту за допомогою апарату неоднорідних семантичних мереж. Реляційно-ситуативна модель заснована на теорії комунікативної граматики Г. А. Золотової.

У теорії комунікативної граматики важливу роль відіграє поняття синтаксеми. Синтаксема є мінімальною синтактико-семантичною одиницею мови, що несе узагальнене, категоричне значення і характеризується взаємодією морфологічних, семантичних і функціональних ознак. Категоріальний зміст синтаксеми називається його значенням. На відміну від ролей, які в лінгвістичних теоріях найчастіше пов'язані з аргументами предикатних слів (наприклад, такі як дієслова) і тісно пов'язані з ними, значення синтаксем в цій теорії в деякій мірі автономні. Це міркування випливає з того, що багато конструкцій в природніх мовах мають зміст навіть у відриві від контексту і предикатного слова. Наприклад, «в лісі», «за лісом», «над лісом» можуть позначати місце дії; «в ліс» – кінцева точка дії; «з лісу» - початкова точка дії; «сокира» є інструментом дії. Крім того, в мові існує безліч виразів, в яких предикатних слів немає. Наприклад, у виразі «будинок в лісі» не вистачає дієслів або інших предикатів слів, зрозуміло, що синтаксема «будинок» має значення якогось об'єкта, а синтаксис «в лісі» – це місце розташування.

Реляційно-ситуативна модель формалізує значення синтаксем речення тексту на природній мові та їх семантичні відношення у вигляді семантичної мережі. Вершини цієї мережі є синтаксеми. Синтаксеми можуть бути двох типів - іменні синтаксеми і предикатні синтаксеми. Значення предикатних синтаксем є зміст

предиката, який вони виражають. Значення номінальних синтаксем у тексті можуть бути представлені як: зв'язки між іменними синтаксемами та синтаксемами предикатів, зв'язки між іменними синтаксемами та службовою вершиною SemRoot, де тип зв'язку визначає значення.

1.3 Висновки. Постановка задачі

Метою даного дослідження є удосконалення якості методів та програмних засобів препроцесингу вхідного текстового коду за допомогою синтаксичного та семантичного аналізу текстів природною мовою шляхом розробки методу семантико- синтаксичного аналізу. Дослідження в першу чергу спрямоване на створення методів аналізу текстів українською мовою. Для цього визначимо, що об'єктом даного дослідження є процес створення семантико-синтаксичних методів аналізу текстів українською мовою, а предметом дослідження є методи і алгоритми визначення рольових структур тверджень, а також методи і алгоритми семантичного розбору.

Для її досягнення необхідно:

1. Провести вивчення методів синтаксичного і семантичного аналізу текстів природною мовою.
2. Розробити метод визначення рольових структур тверджень в текстах українською мовою.
3. Розробити ефективний метод семантико синтаксичного аналізу, в якому інтегровані методи побудови синтаксичних дерев залежностей і визначення рольових структур тверджень.
4. Реалізувати методи семантичного і семантико синтаксичного аналізу. Розробити інтегровану систему семантико синтаксичного аналізу.

5. Провести експериментальні дослідження методів семантичного та семантико синтаксичного аналізу.
6. Розробити методи вирішення прикладних задач, які використовують результати семантичного і семантико синтаксичного аналізу.
7. Провести експериментальне дослідження методів вирішення прикладних задач. Оцінити вплив розроблених методів семантичного та семантико синтаксичного аналізу на якість вирішення цих задач.

2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ

2.1 Інтеграція методів синтаксичного і семантичного аналізу текстів.

Проблема визначення рольових структур тверджень у текстах українською мовою

Завдання визначення рольових структур тверджень, як і більшості інших завдань обчислювальної лінгвістики, має кілька інтерпретацій. Немає єдиної думки про те, які конструкції природної мови слід вважати аргументами предикатних слів, які ролі слід віднести до аргументів: узагальнені, тематичні, специфічні для предикатних слів (PropBank), специфічні для фреймів (FrameNet). Тому для того, щоб сформулювати постановку задачі визначення рольових структур тверджень, необхідно вказати всі перераховані вище аспекти, а також визначити формальну структуру, яка повинна бути в результаті побудована аналізатором. Основною метою його створення був збір і формалізація знань про семантику предикатних слів і синтаксем української та англійської мов в тому вигляді, в якому вони могли б використовуватися в комп'ютерних системах для обробки текстів на природній мові. Семантичний словник зберігає словникові статті, кожен з яких відповідає певній ситуації. В словарній статті знаходяться леми предикатних слів які позначають одну і ту ж ситуацію. Словарна стаття також містить семантичні значення, які можуть мати синтаксеми при заданих предикатних словах, а також признаки, які повинна мати синтаксема, щоб їх були співставлені ті чи інші якості. В якості основних ознак використовуються відмінок, прийменник, а також категоріально-семантичний клас іменника. До категоріально-семантичного класу (КСК) іменників відноситься до їх узагальнене значення. Словникові статті в семантичного словника в деякій мірі чимось схожі з фреймами ресурсу FrameNet – вони визначають список предикатних слів і їх центральні ролі. Таким чином, задача визначення аргументу зводиться до визначення слова-вершини синтаксичної конструкції, значення якої визначається, тому під аргументом можна розуміти як окреме слово з прив'язкою до деякої позиції в синтаксичному дереві речення, а в

якості семантичних аргументів ми розглянемо тільки іменники, займенники, питальні слова і числівники.

Задачу визначення рольових структур тверджень буде розглядати в більш широкому розумінні, тобто ми будемо включати в неї виявлення предикатних слів, виявлення їх аргументів, а також присвоєння семантичних ролей аргументам. На вхід семантичному аналізатору подається дерево синтаксичних залежностей речення, морфологічні характеристики слів, а також категоріально-семантичні класів слів.

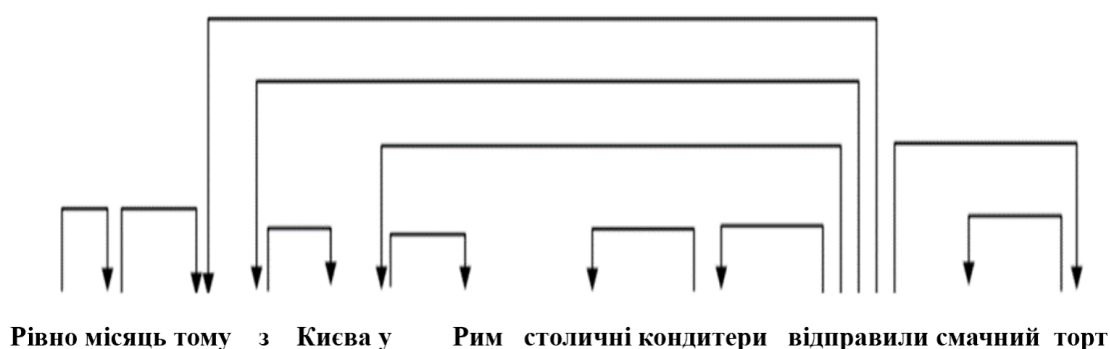


Рисунок 2.1 – Синтаксична структура, яка подається на вхід семантичного аналізатора

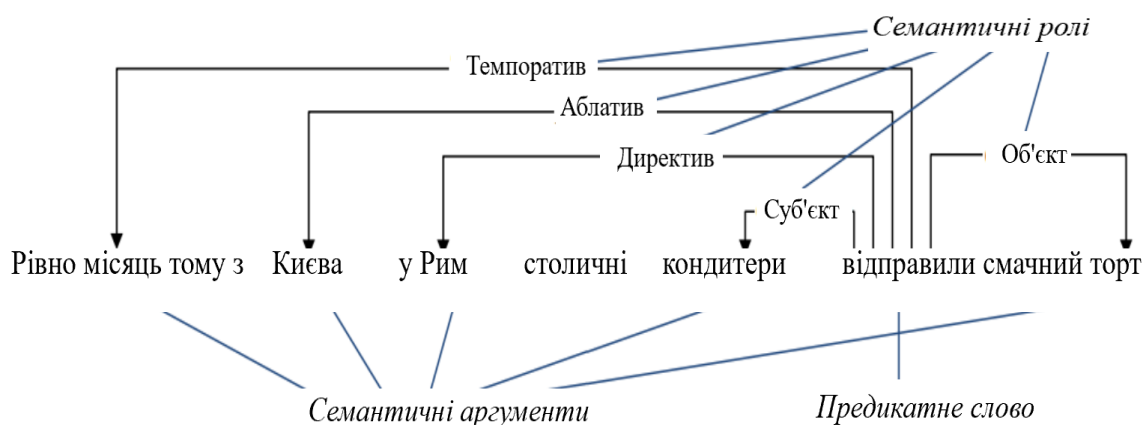


Рисунок 2.2 – Семантична структура – результат роботи семантичного аналізатора

2.2. Метод визначення рольових структур тверджень у текстах українською мовою

Основна концепція даного методу полягає в тому, що, маючи невеликий скінченний набір евристики, детермінований алгоритм і розширюваний семантичний словник, можна побудувати систему визначення рольових структур тверджень в текстах з будь-якої предметної області.

Загальний алгоритм роботи семантичного аналізатора

Алгоритм семантичного аналізу речення можна розділити умовно на п'ять етапів:

- пошук предикатив слів та відповідних словникових статей;
- пошук аргументів предикативних слів;
- призначення семантичних ролей знайденим аргументам відповідно до кожного знайденої словникової статті предикатив слів незалежно від інших аргументів;
- вибір найкращого розподілу семантичних ролей по аргументам для даної словникової статті з урахуванням обмежень унікальності ролей шляхом вирішення задачі оптимізації про призначення;
- вибір найкращої словникової статті предикативних слів та відповідного набору семантичних ролей.

Далі подано короткий опис кожного етапу.

Морфосинтаксичні шаблони використовуються для пошуку предикативних слів у реченні. В основному умови накладаються на частину мови і на положення в синтаксичному дереві. При пошуку предикативних слів-іменників в реченні тексту використовуються лемми предикативних слів з семантичного словника. Після того, як предикатні слова знайдено, для кожного з них встановлюється список відповідних словникових статей у семантичному словнику. Він формується шляхом вибору з розділу предикативних слів рольових структур семантичного словника всіх статей, які містять лему основного слова предикативних слів.

Для кожного знайденого в реченні предикатного слова, здійснюється пошук його семантичних аргументів. Пошук проводиться відповідно до заданого набору евристики, який враховує характеристики предикатних слів, морфологічну характеристику розглянутих в реченні слів, синтаксичну структуру речення. Кожен знайдений аргумент пов'язаний з його вагою - числом від 0 до 1. Вага аргументу відображає впевненість в тому, що знайдене слово дійсно є аргументом предикатного слова.

Для кожної словникової статті предикатного слова і для кожного семантичного аргументу предикатного слова на основі характеристик аргументу та інформації, яка знаходиться в словниковій статті, визначає набір семантичних ролей, які може мати заданий аргумент незалежно від інших аргументів. Ця процедура порівнює відмінок, категоріально-семантичний клас та прийменник аргументу з відповідними функціями в даній словниковій статті для даного предикатного слова. В результаті нечіткого порівняння цих ознак на основі евристики визначається оцінка відповідності аргументу семантичної ролі – вага від 0 до 1.

Після того, як кожному аргументу для даної словникової статті присвоюється набір семантичних ролей і визначається їх вага, проводиться процедура вирішення неоднозначності розподілу ролей по аргументах. У ньому семантичні ролі розподіляються між аргументами так, що кожен семантичний аргумент має не більше однієї ролі для заданого предикатного слова, і кожна роль для заданої статті буде використовуватися не частіше одного разу. При цьому покриття аргументів ролями має бути найкращим відповідно до ваги аргументів і ролей. Для цього використовується алгоритм оптимізації на основі угорського методу, який вирішує задачу про присвоєння.

Після того, як для кожної статті буде знайдено оптимальний розподіл семантичних ролей між аргументами, вибирається найкраща словникова стаття для предикатного слова. В якості оцінки статті використовується значення цільової функції при оптимальному вирішенні відповідної задачі про призначення. В

результаті формується набір семантичних ролей, який найкращим чином охоплює семантичні аргументи для найкращого предикатного слова словникової статті.

Пошук предикатних слів

Предикатні слова шукаються за допомогою набору морфосинтаксичних шаблонів. Предикатні слова може бути як окремими словами, так і конструкціями, що складаються з декількох слів. В останньому випадку визначається головне слово в цій конструкції. Розглянемо набір основних шаблонів для пошуку предикатних слів в текстах українською мовою.

– предикатними словами можуть бути окремі слова наступних частин мови: дієслова, дієприкметники, прислівники, дієприслівник, прикметники і короткі прикметники. При пошуку предикатних слів-іменників їх леми в тексті порівнюються з лемами предикатних слів в семантичному словнику;

– предикатні слова можуть бути виразами двох слів з допоміжним дієсловом виду: <допоміжне дієслово або прислівник> + <дієслово або короткий прикметник (головне слово)>. Приклад: «почали малювати»; «був зламаний»;

– предикатні слова можуть бути словосполученням з двох слів, такими як: <предикатив>+ <дієслово (головне слово) >. Приклад: «можна малювати», «повинно початись»;

– предикатні слова можуть бути конструкцією з трьох слів виду: <предикатив>+ <допоміжне дієслово> + <дієслово або короткий прикметник (головне слово)>. Приклад: «можна було малювати»; «повинено було початись»; «міг бути поламаний»

– деякі конструкції з чотирьох слів також розглядаються як єдине предикатне слово: <предикатив>+ <допоміжне дієслово> + <допоміжне слово> + <дієслово (головне слово)>. Приклад: «можна було почати малювати».

Після виділення предикатного слова визначаються і закріплюються важливі для подальшого семантичного аналізу властивості. Якщо предикатне слово - вираз, що складається з декількох слів, то серед них вибирається основне, яке характеризує всю конструкцію, що позначає предикатне слово.

Розглянемо основні властивості, які визначені для предикатних слів.

– частина мови.

– лема. По лемі шукаються словникові статті в семантичному словнику.

Для більшості предикатних слів лема вилучається з морфологічного розбору головного слова.

– прапорець «оберненість». Семантичний словник також не включає статей для оберненої форми дієслова і статті для коротких прикметників. Для обробки цих випадків часто можна використовувати статтю для оберненої форми дієслова. При цьому признаки семантичних ролей в даному випадку повинні бути модифіковані. Прапорець повернення встановлюється предикатним словом, яке представлено в тексті дієсловами в оберненій формі, короткими прикметниками і дієприкметниками, а також в деяких інших випадках.

Лема предикатного слова визначає список його словникових статей у семантичному словнику. У цьому випадку одна лема часто відповідає кільком словниковим статтям. Це пов'язано із змістом омонімії предикатного слова. У прикладах «Вино підходить до вечері» і «Гості підійшли до ресторану» предикатне слово «підхід» має різне значення. У семантичному словнику ці два значення відповідають двом різним словниковим записам, які вказують на різні рольові структури.

В результаті формується список предикатних слів речення, для яких фіксуються вищевказані характеристики і визначаються списки їх можливих словникових статей.

Пошук семантичних аргументів. Для кожного знайденого предикатного слова визначається множина його семантичних аргументів у реченні.

Нехай задано деяке речення і в ньому є предикатне слово. Всі слова в реченні переглядаються зліва направо і перевіряються на відповідність ряду умов. Перш за все, перевіряється, чи слово має всі морфо-лексичні ознаки, щоб вважати його семантичним аргументом. Відповідно до постановки задачі семантичні аргументи можуть бути іменниками, займенниками, питальними словами і чисельниками.

Частина мови слова відома з морфологічного аналізу, перевірка на те, що слово є запитальним виконується за допомогою окремого словника.

Далі береться до уваги, яке місце займає слово в синтаксичній структурі речення. Було визначено ряд синтаксичними конструкцій, що вказують на те, що слово є семантичним аргументом.

- аргумент синтаксично безпосередньо підпорядкований предикатному слову.

- аргумент синтаксично підпорядкований прийменнику, і цей прийменник підпорядкований предикатному слову. Також враховується, що прийменники можуть складатися з декількох слів: «в результаті», «разом з», «за рахунок» і т. д. Складені прийменники розпізнаються в реченні окремою процедурою, яка використовує словник подібних конструкцій. У такому випадку, коли дане слово підпорядковане одному зі слів складного прийменника, перевіряється, що вершина синтаксичної групи такого прийменника підпорядкована предикатному слову.

- у випадку, коли предикатним словом є прислівник, в якості семантичного аргументу виділяється також і слово, визначене прислівником.

- у випадку, коли предикатним словом є дієприслівник, який в якості семантичного аргументу виділяється і підмет простого речення, обставиною якого є дієприслівниковий оборот з відповідним предикатним словом

- у випадку однорідних предикатних слів аргументи ліворуч від першого однорідного предикатного слова, стають загальними семантичними аргументами для всіх предикатних слів однорідної групи.

- якщо предикатне слово є складеним, то перераховані правила застосовуються до кожного слова складеного предикатного слова.

Призначення ролей семантичних аргументів. Припустимо, є предикатне слово, множина його семантичних аргументів, задана одна словникова стаття предикатних слів.

Процедура призначення ролей семантичних аргументів починається з призначення «основних» ролей. «Основна» роль – це та роль предикатного слова, яка найчастіше зустрічається в ньому і без яких опис ситуації, визначеної предикатним словом, часто семантично неповна. Присвоєння основних ролей аргументам здійснюється в основному на основі ознак, що містяться в розділі рольових структур предикатних слів семантичного словника.

Якщо будь-який семантичний аргумент є питальним словом, а на попередньому етапі йому не була призначена роль, то починається окрема процедура присвоєння ролей питальним словам, яка виконується на основі інформації, що міститься в розділі питальних слів семантичного словника.

Якщо роль семантичного аргументу не визначено після попередніх кроків, то запускається процедура визначення периферійних або «необов'язкових» ролей, яка виконується на основі інформації, що міститься в розділі периферійних ролей.

Призначення основних ролей аргументам

Нехай це буде задане предикатне слово, одна з його словникових статей і один семантичний аргумент.

Всі ролі даної словникової статті розглядаються в циклі і їх ознаки нечітко порівнюються зі ознаками семантичного аргументу. Результатом порівняння є число від 0 до 1, яке ми назвемо вагою семантичної ролі. Вага – середнє геометричне двох коефіцієнтів (від 0 до 1): близькість за категоріально семантичним класом і близькість за граматичними ознаками. Обидва коефіцієнти визначаються за допомогою набору евристик.

При визначенні коефіцієнта близькості за категоріально семантичним класом аналізатор послідовно порівнює всі категоріально-семантичні класи, зазначені для заданої ролі в словниковій статті, з категоріально-семантичним класом аргументу і для кожного з них визначає оцінку подібності. При цьому збіг одних категоріально-семантичних класів вважається більш значущим, ніж збіг інших. На цю оцінку також накладається штраф, якщо роль у словниковій статті має кілька категоріально-семантичних класів або роль допускає будь-який клас. Якщо значення аргументу категоріально-семантичного класу «не визначено», то оцінка

набуває невелике значення, більше нуля. В якості остаточного коефіцієнта близькості для категоріально-семантичного класу вибирається максимальна оцінка.

Визначення коефіцієнта близькості за граматичними ознаками здійснюється шляхом порівняння пари ознак (прийменник; відмінок) семантичного аргументу з признаками, заданими для даної ролі в словниковій статті. Як правило, словникова стаття вказує не на одну, а декілька підходящих пар ознак, придатних для даної семантичної ролі. Коефіцієнти близькості розраховуються окремо для кожної пари, а потім з них вибирається максимальний коефіцієнт і відповідна пара ознак зі словника.

Коефіцієнт близькості за граматичними ознаками розраховуються наступним чином. Якщо прийменник аргументу не збігається з прийменником ролі, значення коефіцієнта дорівнює 0. В іншому випадку розраховується близькість за відмінками. Перед обчисленням оцінки близькості за відмінками перевіряється дане предикатне слово, чи встановлено знак «зворотність». Якщо цей признак встановлено, то відмінки зі словникової статті для даної ролі змінюються наступним чином. Якщо немає прийменника для даної ролі, то називний відмінок замінюється орудним відмінком, а знахідний відмінок називним відмінком. Деякі перетворення накладають штраф на коефіцієнт близькості по граматичними ознаками

Через відмінкову омонімію у аргумента може бути декілька варіантів відмінків. Якщо аргумент має один варіант відмінка, який збігається з відмінком семантичної ролі, то оцінка наближення по відмінкам - 1. Якщо у аргумента декілька варіантів відмінка і хоча б один з них збігається з відмінком ролі в словниковій статті, то оцінка не дорівнює нулю (при налаштуванні аналізатора встановлюється конкретне значення (0; 1]). В іншому випадку оцінка близькості за відмінками дорівнює нулю.

Після обробки всіх ролей, присутніх в словниковій статті, формується список можливих ролей семантичного аргументу і їх ваги.

Призначте семантичні ролі аргументам, вираженим питальними словами.

У цій процедурі ролі семантичних аргументів визначаються вираженими питальними словами: «хто», «що», «коли», «де», «куди» і т. д. Обробляються тільки ті семантичні аргументи, роль яких не була визначена в результаті процедури визначення основних ролей. Семантичні ролі присвоюються питальним словам за допомогою окремого розділу семантичного словника. Цей розділ містить словникові статті, які вказують, які ролі можуть мати питальні слова і які морфологічні особливості вони повинні мати для цього.

По лемі питального слова в словнику шукають відповідну статтю. Якщо питальним словом є прислівник або сполучник, то всі ролі з словникової статті додаються до списку можливих семантичних ролей семантичного аргументу. Якщо ні, то перевіряються морфологічні ознаки питального слова. Морфологічні ознаки перевіряються так само, як і при призначенні основних ролей аргументам.

У остаточному списку можливих семантичних ролей питальних слів залишаються тільки ті ролі, які є в словниковій статті предикатного слова і ролі, які входять до списку периферійних семантичних ролей: локатив, темпоратив, інструментів, директив, казуатів і т. д. Вплив ролей питальних слів занижені в порівнянні з впливом основних ролей інших семантичних аргументів.

Призначення периферійних семантичних ролей аргументам

Процедура призначення периферійних ролей обробляє лише ті семантичні аргументи, яким ще не призначено жодної ролі після попередніх процедур. Периферичні семантичні ролі призначаються аргументам за допомогою розділу периферійних ролей семантичного словника.

Для заданого семантичного аргументу формується множина наборів ознак: трійок (категоріально-семантичні класи, відмінок, прийменник). Ці трійки шукаються в розділі семантичного словника для периферійних ролей, і з яких вибираються відповідні їм семантичні ролі. Для кожної вибраної ролі перевіряється, чи присутня вона у словарній статті предикатного слова аргументу. Якщо така роль присутня в статті предикатного слова, то вона відкидається. В іншому випадку роль додає аргумент до списку можливих семантичних ролей, його вага знижується в порівнянні з вагою головних ролей

Вирішення неоднозначності призначення семантичних ролей аргументам

В результаті всіх попередніх етапів формується список з декількох можливих семантичних ролей для кожного семантичного аргументу. Більш того, різні семантичні аргументи можуть повторювати ролі в цих списках. Однак, згідно з принципами моделі семантики тексту, заснованої на рольовій структурі речення, з одного боку, кожен семантичний аргумент може мати тільки одну роль, а з іншого боку, одна і та ж роль не може бути в декількох семантичних аргументах.

Проблема виникає для оптимального розподілу семантичних ролей за аргументами, так щоб оцінка рольової структури для заданого предикатного слова була максимальною. Пропонується розглядати цю проблему як задачу присвоєнь, що в свою чергу є проблемою лінійного цілочисельного програмування. Загалом, постановка задачі присвоєння у вигляді завдання лінійного цілочисельного програмування виглядає наступним чином

$$\begin{aligned} & \operatorname{argmin}_{x_{ij}} \sum_{i,j=1}^n C(i,j)x_{ij} \\ & \sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n \\ & \sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n \\ & x_{ij} \in \{0,1\}, i, j = 1, \dots, n \end{aligned}$$

Тут x_{ij} є змінними, $C(i,j)$ - функція вартості.

Представимо математичну постановку задачі про присвоєння семантичних ролей семантичним аргументам. Нехай буде впорядкована множина A - семантичні аргументи. Нехай R_i буде множиною можливих ролей аргументу $a_i \in A$. Нехай R це впорядкована множина всіх унікальних ролей, які можливо встановити аргументам A . Оскільки в цілому $|A| \neq |R|$, задамо $n = \max\{|A|, |R|\}$ і дві множини $A^* \text{ і } R^*$, такі, що $|A^*| = |R^*| = n$; $a_i \in A$, $a_i^* \in A^*$, $a_i = a_i^*$, $i = 1, \dots, |A|$; $r_j \in R$, $r_j^* \in R^*$, $r_j = r_j^*$, $j = 1, \dots, |R|$. Функцію вартості $C(a, r)$ задамо як добуток аргументу w^a , визначеного на етапі пошуку семантичних аргументів, що відповідає аргументу семантичної ролі

w^r . Більш того, якщо в списку можливих семантичних ролей даного аргументу $a_k \in A$ роль $r_m \in R$ відсутня ($r_m \notin R_k$), то вартість дорівнює нулю $C(a_k, r_m) = 0$. Нехай для всіх аргументів $a_i^* \in A^*$, які не входять в початкову множину A ($a_i^*, i > |A|$), функція вартості дорівнює нулю. Нехай для всіх ролей $r_j^* \in R^*$, які не входять в початкову множину R ($r_j^*, j > |R|$), функція вартості дорівнює нулю. Запишемо повний вираз для функції вартості:

$$C(a_i, r_j) = \begin{cases} w_i^a * w_j^r, & i \leq |A|, j \leq |R|, r_j \in R_i \\ 0, & r_j \notin R_i \\ 0, & i > |A| \text{ или } j > |R| \end{cases}$$

Задамо змінні $x_{ij} \in \{0, 1\}$, де $x_{ij} = 1$ означає, що аргументу a_i присвоюється роль r_j . Оцінка семантичної структури представлена як лінійний функціонал $Sem = \sum_{i,j=1}^n C(a_i, r_j) x_{ij}$. Тоді завдання присвоєння семантичних ролей семантичним аргументам можна представити у наступному вигляді.

$$\begin{aligned} \operatorname{argmax}_{x_{ij}} Sem &= \operatorname{argmax}_{x_{ij}} \sum_{i,j=1}^n C(a_i, r_j) x_{ij} \\ C(a_i, r_j) &= \begin{cases} w_i^a * w_j^r, & i \leq |A|, j \leq |R|, r_j \in R_i \\ 0, & r_j \notin R_i \\ 0, & i > |A| \text{ или } j > |R| \end{cases} \\ \sum_{i=1}^n x_{ij} &= 1, j = 1, \dots, n \\ \sum_{j=1}^n x_{ij} &= 1, i = 1, \dots, n \\ x_{ij} &\in \{0, 1\}, i, j = 1, \dots, n \end{aligned}$$

Представлена задача вирішується за допомогою угорського методу. Відповідно до отриманого рішення аргументам присвоюються ролі з вказаної словникової статті предикатних слів. Отримана структура є оптимальною для

обраних оцінок ролей і аргументів. Значення цільової функції визначає, наскільки добре визначена словника стаття підходить для заданого предикатного слова і його аргументу. Воно використовується для вибору оптимальної словникової статті і відповідній їй рольовій структурі предикатного слова.

2.3. Метод семантико-синтаксичного аналізу текстів

Семантичні ролі можуть бути додатковим джерелом інформації для синтаксичного аналізу. Вони несуть знання про сумісність предикатних слів з аргументами. Рольова структура предикатного слова визначає набір «обов'язкових» аргументів, які повинно прикріпити до себе предикатне слово. Крім того, в процесі встановлення семантичних ролей вирішується проблема вирішення семантичної омонімії предикатних слів (вибір словникової статті). На рисунку 2.3 показаний приклад, в якому, навпаки, варіант розбору, в якому іменна група «без наслідків» підпорядковується слову «криза», семантично правильний, але варіант, в якому ця група пов'язана з дієсловом «буде відображений» - ні.

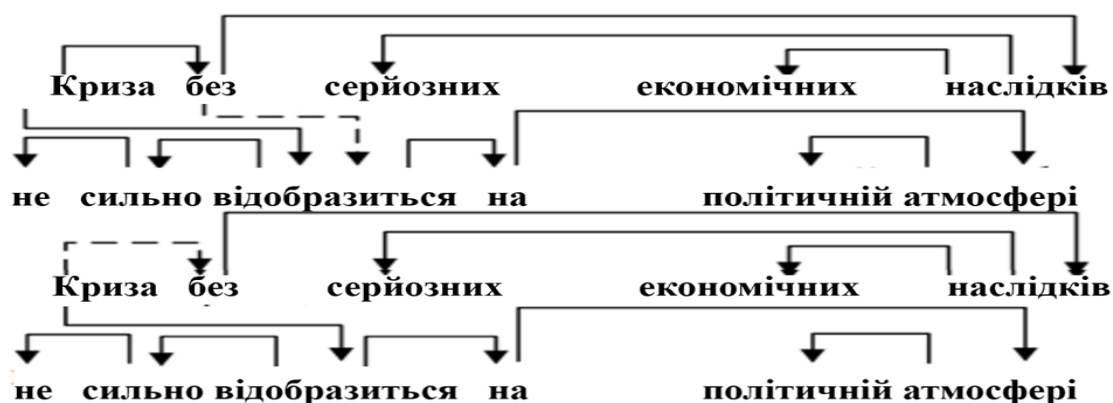


Рисунок 2.3 – Приклад того, як синтаксичний аналізатор може розставити відносини у реченні «Криза без серйозних економічних наслідків не сильно вплине на політичну атмосферу».

Даний розділ містить метод, який інтегрує методи побудови синтаксичних дерев залежностей і визначення рольових структур тверджень. Завдяки знанням про рольову структури речення, отриманої від семантичного аналізатора, метод коректує початкове синтаксичне дерево залежностей. Виправлення в синтаксичному дереві, в свою чергу, дозволяють будувати більш повні рольові структури. Таким чином, метод дозволяє покращити якість синтаксичного і семантичного аналізу.

Загальний опис методу семантико-синтаксичного аналізу. Передбачається, що до семантико-синтаксичного аналізу була проведена токенізація тексту, і текст розбитий на речення, та проведений його морфологічний аналіз, включаючи визначення морфологічних ознак слів, лематизацію і омонімізацію. Семантико-синтаксичний аналіз виконується для кожного речення окремо.

Першим кроком є виконання початкового синтаксичного аналізу речення. Віно може бути виконаний будь-яким аналізатором, який будує дерево залежності синтаксису. Деревя, побудовані в результаті первинного аналізу, додатково коригуються за рахунок інформації, отриманої при визначенні рольових структур тверджень.

На наступному кроці речення шукає предикатні слова. Предикатні слова, виражені іменниками або дієприкметниками, відкидаються.

Потім для кожного знайденого предикатного слова, виходячи із синтаксичної структури речення визначаються його семантичні аргументи. Пошук семантичних аргументів предикатного слова здійснюється за допомогою аналізатора на основі «сильних» правил, тобто такого аналізатора, який шукає аргументи у відповідності до побудованого синтаксичного дерева залежностей. Якщо в синтаксичному дереві є помилка, аналізатор не знайде семантичний аргумент. Таким чином, набір семантичних аргументів, виділених цим аналізатором, досить точний, але може бути неповним. Далі цей набір семантичних аргументів буде називатися «базовим».

Потім ми шукаємо конструкції, які можуть бути семантичними аргументами, але через потенційні помилки в синтаксичному дереві вони не були ідентифіковані.

Пошук проводиться за допомогою аналізатора на основі «слабких» правил. Цей аналізатор не накладає обмежень на синтаксичну зв'язність предикатного слова і семантичного аргументу. Він передбачає, що синтаксичні зв'язки можуть бути встановлені не точно і покладається на інші признаки. Конструкції, виділені цим аналізатором, будуть називатися «можливими» семантичними аргументами. Набір можливих семантичних аргументів є досить повний, але неточний. Як правило, він включає в себе набір основних аргументів.

Серед можливих семантичних аргументів, які не є основними, шукаються «додаткові» аргументи. Додатковими аргументами будуть називатися можливі аргументи, наявність яких в реченні допускається рольовою структурою предикатного слова, і вони мають для цього відповідні особливості. Додаткові аргументи фільтруються за допомогою набору евристик.

Після того, як будуть знайдені всі додаткові аргументи всіх предикатних слів речення, для кожного додаткового аргументу визначаються виправлення синтаксичного дерева (додавання/видалення зв'язків), які повинні бути зроблені для того, щоб синтаксична структура відповідала новій семантичній структурі речення, з додаванням семантичних аргументів. Виправлення також проходять ряд перевірок з використанням евристик.

В результаті виправлене дерево синтаксису повторно подається на семантичний аналіз.

Пошук можливих семантичних аргументів. Методи знаходження можливих семантичних аргументів схожі між собою, але даний метод накладає менш суворі обмеження на синтаксичну структуру речення і не вимагає, щоб можливі аргументи були синтаксично пов'язані з предикатними словами. Однак для того, щоб відфільтрувати найменш підходящі варіанти, він вводить ряд додаткових обмежень на аргументи.

Перш за все, формуються ліва і права межі пошуку додаткових аргументів для даного предикату слова. Межі визначаються відповідно до невеликого набору правил, які призначені для фільтрації слів, які потрапляють в область, де з лінгвістичних причин не можуть бути знайдені аргументи, синтаксично пов'язані з

даним предикатним словом, або в цій області потенційно можуть бути аргументи, синтаксично пов'язані з іншими предикатними словами. При побудові меж враховуються розділові знаки і сполучники, що розділяють однорідні предикатні слова.

Далі в рамках побудованих меж виділяються всі слова, що задовольняють ознаки іменних аргументів: іменники, особові займенники, займенники «що», «один», питальні слова і т.д. Вибрані слова проходять більш складну серію перевірок. Серед найбільш значущих перевірок слід зазначити наступне:

- перевірка того, що можливий семантичний аргумент при заданому предикатному слові не є основним аргументом в іншому предикатному слові

- перевірка структури з дужками: можливий аргумент повинен бути на тому ж рівні структури з дужками, що і зазначене предикатне слово. Схожа перевірка здійснюється і для структури з лапками.

- перевірка того, що можливий аргумент є вершиною складеної іменної сутності. Наприклад, у конструкції ППП, проаналізованій як $\Pi_1 \rightarrow I \rightarrow \Pi$, слово Π_1 буде виділено як єдиний додатковий аргумент.

- переконатись, що можливий аргумент не є частиною сталого виразу.

Текст виділяє вирази, які вказують на час, період часу, періодичність, вартість, значення за певний період часу, складені числа, процентні вирази, одиниці вимірювання тощо.

- перевірка того, що можливий аргумент не є однорідним членом речення, який підпорядкований іншому однорідному члену. Однорідні слова можуть позначати той же семантичний аргумент, тоді як підпорядковане однорідне слово синтаксично не залежить від предикатного слова, тому воно не розглядається як додатковий аргумент.

Після цього з множини можливих аргументів відсіюються ті аргументи, які входять в множину основних аргументів шляхом простого віднімання множин. В результаті операції різниці побудовано багато можливих аргументів, які

потенційно можуть бути семантичними аргументами предикатного слова, але в початковому синтаксичному дереві немає необхідних синтаксичних зв'язків.

Визначення додаткових семантичних аргументів

Додаткові семантичні аргументи відрізняються від можливих аргументів тим, що їх наявність підтверджується можливістю включення їх в рольову структуру предикатного слова. Для того щоб відрізнити додаткові аргументи від набору множини аргументів, аналізатор намагається призначити семантичні ролі можливим аргументам, які не суперечать ролям «основних» аргументів.

Для цього аналізатор спочатку призначає семантичні ролі «основним» аргументам зазначеного предикатного слова. Потім кожний можливий аргумент додається до набору основних аргументів по черзі, і аналізатор намагається призначити ролі розширеному набору аргументів. Якщо доданому можливому аргументу не призначить роль, його відхиляють. Якщо цьому аргументу призначено роль, перевіряється, чи належить роль аргументу з основного нерозширеного набору. Якщо ні, то до набору додаткових аргументів додається можливий аргумент, інакше можливий аргумент буде відхилено.

Після аналізу всіх можливих аргументів до набору основних аргументів додаються виділені додаткові аргументи, формується загальний набір семантичних аргументів. Процедура семантичного аналізу запускається заново на цьому наборі. Ті додаткові аргументи, які не отримали семантичну роль, відкидаються. Це вирішує проблему з додатковими аргументами, ролі яких дублюються.

В результаті цієї процедури будується набір додаткових аргументів, їм призначаються ролі, і ці ролі не суперечать ролям аргументів з основного набору.

Налаштування синтаксичного дерева залежностей. Для кожного додаткового аргументу, знайденого на попередньому кроці, робиться спроба налаштувати синтаксичне дерево таким чином, щоб цей аргумент можна було вибрати як одне з правил алгоритму пошуку основних аргументів. При цьому виправлена синтаксична структура повинна залишатися деревом: кожен вузол має не більше одного батьківського, і циклів немає.

Щоб виправити дерево, зазвичай потрібно видалити один або декілька синтаксичних зв'язків та додати один зв'язок. Тут можуть виникнути наступні варіанти:

– слово, призначене як додатковий аргумент, не підпорядковане прийменнику або конструкції, що позначає прийменник. У цьому випадку він повторно перепідпорядковується безпосередньо предикатному слову, а синтаксичний зв'язок між словом і його початковим батьком видаляється. (рис. 2.4.А)

– слово, призначене як додатковий аргумент, синтаксично підпорядковане прийменнику або конструкції, що позначає складений прийменник. При цьому перепідпорядковується не слово, а прийменник. У разі складеного прийменника перепідпорядковується його синтаксичну вершина.(рисунок 2.4, Б)).

– предикатне слово є дієприкметником. Тоді окремий алгоритм визначає, чи може додатковий аргумент бути визначальним словом. Він перевіряє узгодженість дієприкметників і додатковий аргумент у роді, числі та відмінку, а також визначає, чи має дієприкметники узгодженого батька як альтернативу. Якщо аргумент узгоджений з дієприкметником, а дієприкметник немає альтернативного узгодженого батька, то дієприкметник переузгоджується до цього аргументу (рисунок 2.4, В).

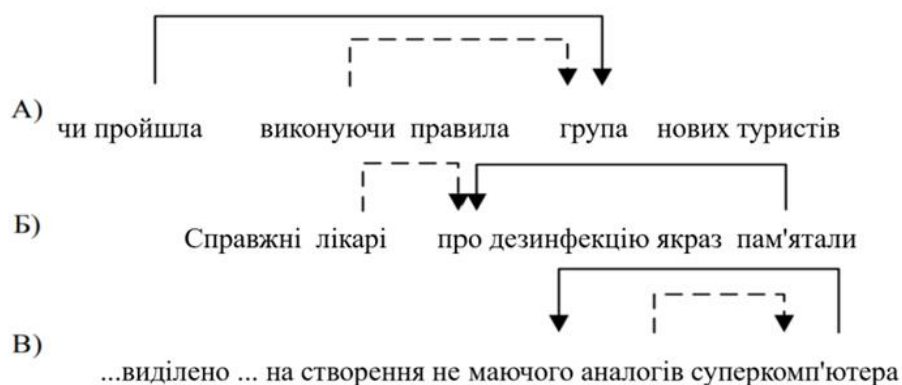


Рисунок 2.4 – Приклади способів налаштування дерева синтаксису.

Пунктирна стрілка вказує на синтаксичний зв'язок, помилково встановлений початковим аналізатором, який видаляється системою семантико-синтаксичного аналізу; суцільна стрілка вказує на новий зв'язок, додану системою семантико-синтаксичного аналізу

Після того, як ви визначили перетворення, які потрібно застосувати до синтаксичного дерева, щоб відрегулювати його, перевіряється, щоб перетворення не призвели до циклу, який проходить через додатковий аргумент і предикатне слово. Якщо такий цикл формується, його можна розірвати двома способами:

- якщо предикатне слово є дієприкметником, а додатковий аргумент є визначальним словом, то в циклі видаляється зв'язок, вершиною якого є предикатне слово.
- в іншому випадку в циклі видаляється зв'язок, вершиною якого є аргумент.

Далі перевіряється, чи є доданий зв'язок у виправленому синтаксичному дереві проєктивним.

Перед фіксацією виправлень виконується ще одна перевірка перетворень синтаксичного дерева для мінімізації помилкових коригувань. Для цієї перевірки розроблено два методи: за допомогою використання статистично-евристичного критерію та за допомогою ансамблю бінарних класифікаторів.

Розглянемо перший метод перевірки виправлень за допомогою статистично-евристичного критерію. Припустимо, є два іменники, між якими безпосередньо встановлено синтаксичний зв'язок, наприклад, «розробка додатків», або встановлюється з'єднання через прийменник, наприклад, «згода на розробку». Відповідно до навчального корпусу розраховується частота співучасті лем головного іменника з загальною кількістю трьох ознак залежного іменника: відмінок, прийменник і КСК. Невелика частині навчального корпусу визначає поріг. Якщо при аналізі початковий синтаксичний аналізатор встановлює синтаксичний зв'язок між двома іменниками, а частота при якій лема і головне слово зустрічаються разом вище цього порогу, то корекція в синтаксичному дереві не виконується.

Другий метод перевірки виправлень використовує набір з трьох бінарних класифікаторів на основі машинного навчання. Перший класифікатор C_d оцінює вагу віддаленого зв'язку між додатковим аргументом і його початковим батьківським. Другий класифікатор C_a оцінює вагу доданого зв'язку між аргументом і предикатним словом. Третій класифікатор C_f бере як вхідні значення цільових функцій з перших двох і приймає остаточне рішення фіксувати чи ні зміни в синтаксичному дереві.

Припустимо, ϵ слово - додатковий аргумент, головне слово, від якого синтаксично залежить аргумент у вихідному синтаксичному дереві (цей зв'язок видаляється), і предикатне слово, якому необхідно перепідпорядкувати аргумент. Розглянемо набори ознак для кожного класифікатора («+» означає конкатенацію ознак).

В класифікаторі C_d використовуються наступні признаки:

- 1) Частина мови головного слова + частина мови аргументу.
- 2) Частина мови головного слова + відмінок аргументу + рід аргументу + число аргументу + відмінок головного слова + рід головного слова + число головного слова.
- 3) Аргумент прийменника + відмінок аргументу + частина мови головного слова.
- 4) Морфологічні признаки головного слова + морфологічна особливості аргументу + частина мови головного слова.
- 5) Якщо головне слово – іменник: лема головного слова + відмінок аргументу + прийменник аргументу + КСК аргументу.
- 6) Якщо головне слово має батька у початковому синтаксичному дереві: частина мови батька головного слова + частина мови аргументу + частина мови головного слова.
- 7) Якщо батько головного слова має батька в початковому синтаксичному дереві: частина мови головного слова + частина мови батька головного слова + частина мови батька головного слова.

8) Якщо аргумент – не є останнім словом в реченні: частина мови слова, наступного за аргументом, частина мови аргументу, частина мови головного слова.

9) Положення аргументу щодо головного слова + прийменник аргументу + частина мови головного слова.

10) Відстань в токенах між головним словом і аргументом.

Класифікатор C_a використовує такі признаки:

1) 1)Лема предикатного слова + КСК аргумент + відмінок аргументу + прийменник аргументу + позиція аргументу відповідно предикатного слова + прапор повернення предикатного слова.

2) Відстань між аргументом і предикатним словом в токенах.

3) Випадок аргументу + КСК аргументу + позиція аргументу щодо предикатного слова + прийменник аргументу.

4) Якщо у предикатного слова є батько: частина мови батьківського предикатного слова.

5) Прапор, який сигналізує про те, що аргумент і предикатне слово знаходяться в одній клаузі.

Класифікатор C_f використовує такі признаки:

1) Значення цільової функції класифікатора C_d

2) Значення цільової функції класифікатора C_a .

3) Семантична роль аргументу.

4) Вага семантичної ролі.

5) Відмінок аргументу.

Ознакові простори класифікаторів C_d і C_a дуже великі. Тому заради швидкості семантико-синтаксичного аналізу був використаний один з найшвидших дискримінантних алгоритмів класифікації – метод опорних векторів без ядра (з лінійним ядром). У класифікаторі C_f є набагато менше ознак. Проведені експерименти показали, що кращу якість перевірки виправлень дає метод опорних векторів з радіальним ядром.

При навчанні класифікатора C_d позитивні і негативні приклади отримували з корпусу наступним чином:

- в якості позитивних прикладів розглядалися пари слів, пов'язаних синтаксичним зв'язком, де залежне слово має риси семантичного аргументу, а головне слово не виділяється як предикатне слово.

- як негативні приклади розглядалися пари слів, між якими немає синтаксичних зв'язків, в той час як один з них має признаки семантичного аргументу, і зв'язок, який може бути зроблений між ними, є проєктивним щодо всіх зв'язків синтаксичного дерева, за винятком зв'язку, в якому залежне слово, що має характеристики аргументу.

При навчанні класифікатора C_a позитивні і негативні приклади витягувалися з корпусу наступним чином:

- як позитивні приклади розглядалися пари слів, між якими існує синтаксичний зв'язок, при цьому одне з них є предикатним словом, а інший має признаки семантичного аргументу.

- як негативні приклади розглядалися пари слів, між якими немає синтаксичного зв'язку, при цьому одне з них є предикатним словом, а інше має признаки семантичного аргументу, і зв'язок, який може бути проведений між ними, є проєктивним щодо всіх зв'язків синтаксичного дерева, за винятком зв'язку, в якому залежне слово, має признаки аргументу.

Класифікатор C_f був навчений на тих випадках корекції синтаксичного дерева, які пройшли всі попередні фільтри, включаючи перевірку на проєктивність результуючого дерева.

Після того, як всі перевірки пройдені, виправлення фіксується в синтаксичному дереві, і алгоритм приступає до обробки наступного додаткового аргументу.

2.4 Висновки

Отже, у даному розділі розроблено нові методи семантичного та семантико-синтаксичного аналізу текстів природною мовою.

Метод семантичного аналізу використовує семантичний словник як основне джерело знань про предикатні слова та їх рольові структури. Морфо-лексичні шаблони використовуються для пошуку предикатів слів. Для пошуку семантичних аргументів предикатного слова використовуються правила, які враховують морфологічні особливості слів і синтаксичну структуру речення. Метод присвоєння ролей аргументам виконує нечітке порівняння особливостей аргументів з особливостями ролей зі статей семантичного словника для заданого предиката слова. Вирішення неоднозначності розподілу ролей по аргументам здійснюється шляхом вирішення задачі оптимізації лінійного цілого програмування угорським методом.

Метод семантико-синтаксичного аналізу використовує інформацію, отриману в результаті семантичного аналізу, для корекції дерева синтаксису. Цей метод шукає слова, які потенційно можуть бути семантичними аргументами, але через помилки в дереві синтаксису речення вони не виявляються звичайним способом, і перевіряється можливість включення їх в рольову структуру предикатного слова. Якщо рольова структура слова предикату дозволяє отримати додатковий аргумент, дерево синтаксису налаштовується відповідно до нової структури ролей. При коригуванні дерева виключаються потенційно виникаючі цикли, перевіряється проектність доданих зв'язків, а також здійснюється перевірка перетворення або за допомогою статистично-евристичного критерія, або за допомогою ансамблю бінарних класифікаторів на основі машинного навчання. На виправленому синтаксичному дереві по новому проводиться семантичний аналіз .

3 АЛГОРИТМ ТА ТЕХНОЛОГІЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

3.1 Використання семантичного та семантико-синтаксичного аналізу для побудови семантичної мережі реляційно–ситуаційної моделі

Модель семантики тексту на основі ролей є дуже ефективним інструментом у вирішенні багатьох прикладних задач автоматичної обробки тексту природною мовою. Однак, як зазначалося раніше, вона досить поверхнева, багато змістовних аспектів, виражених в тексті, залишаються поза його сферою застосування. Більш глибоке представлення семантики тексту відображається в реляційно–ситуаційній моделі тексту. У ній синтаксеми і зв'язки між ними утворюють неоднорідну семантична мережу. Реляційно–ситуаційна модель, крім інформації про семантичні ролі, також враховує значення вільних синтаксем і семантичних відношень. Розглянемо метод встановлення семантичних відносин, який необхідний для побудови повноцінної реляційно–ситуаційної моделі.

Семантичні відношення встановлюються між аргументами предикатних слів. Метод використовує словник семантичних відношень, в якому предикатним словам з семантичного словника співставляється множина семантичних відношень. Кожне семантичне відношення характеризується набором пар семантичних ролей. Ці пари визначають, які ролі повинні мати аргументи для участі в певному семантичному відношенні. У таблиці 3.1 наведено приклад запису в словник семантичних зв'язків для ситуації, описаної предикатним словом «викликати» в прикладі «помилка лікарів викликала епідемію». У роботі використовувався словник семантичних зв'язків, в якому було вказано 35 унікальних семантичних відношень для близько трьох тисяч предикатних слів.

Для кожного речення тексту семантичні відношення визначаються окремо. Припустимо, що існує структура речення предикатно-аргументна, побудована в результаті семантичного або семантико-синтаксичного аналізу.

Таблиця 3.1 – Приклад статті в словнику семантичних відношень для предикатного слова «викликати»

| CO | Роль головного аргументу CO | Роль залежного аргументу CO |
|------|-----------------------------|-----------------------------|
| CAUS | Каузатив | результатив |
| SIT | Ситуатив | каузатив |

Кожне предикатне слово і його аргументи розглядаються окремо. Зі словника семантичних відношень вилучається множина відношень, що відповідають ситуації, зазначеній заданим предикатним словом. Для кожного відношення шукається пара аргументів, яким призначено ролі, які відповідають ролям в словниковій статті для даної ситуації. Якщо пара знайдена, то семантичне відношення додається в семантичну мережу пропозиції.

Семантичні відношення, в побудованій таким чином семантичній мережі, дозволяють знаходити і нормалізувати відношення між концептами предметної області, які відображені в тексті. На рисунку 3.1 показані побудовані рольові структури двох речень і виділені в них семантичні зв'язки (зелені пунктирні стрілки, напис на стрілці вказує тип зв'язку). У цих прикладах, хоча аргументи «епідемія» і «помилка» мають різну семантичну роль, а самі аргументи відносяться до різних предикатних слів (відповідно, «предикат» і «казуатив- на рисунку А»), при предикатному слові «статися»; відповідно «результив» і «казуатив» з предикатним словом «привести» на рисунку Б)), між ними встановлюється той же семантичний зв'язок «CAUS», що позначає причинно-наслідковий зв'язок.

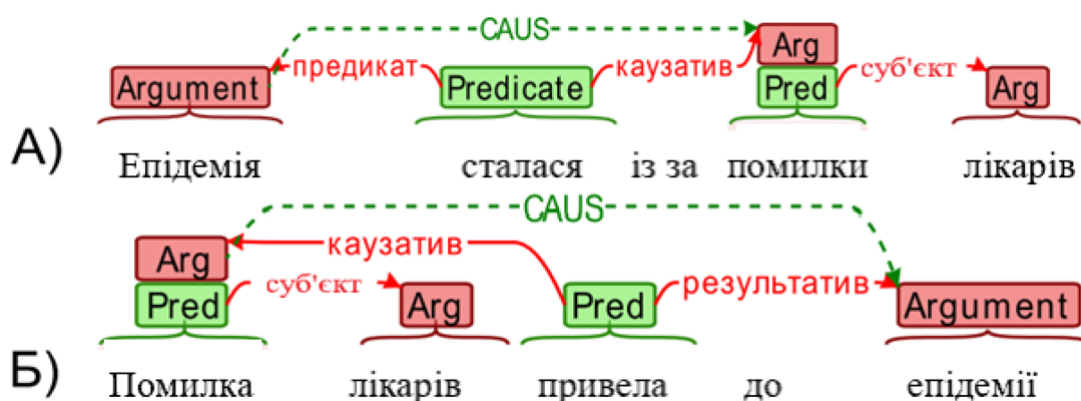


Рисунок 3.1 – Приклад, в якому аргументи мають різні семантичні ролі, але між ними встановлюється однаковий семантичний зв'язок

Семантична мережа може бути використана для вирішення багатьох практичних завдань у сфері пошуку інформації та отримання інформації.

3.2. Застосування семантичного та семантико–синтаксичного аналізу для запитально–відповідального пошуку

Завдання пошуку запитань–відповідей полягає в тому, щоб дати користувачеві відповідь на питання, поставлене природною мовою. Існує безліч формулювань цієї проблеми і типів систем запитань–відповідей. Відповідь можна знайти як в структурованих базах знань, так і в колекціях неструктурованих текстових документів. У цьому розділі розглядається проблема пошуку запитань–відповідей в метапошуковій системі.

Багато інтернет-пошукових систем або не мають питально–відповідального модуля пошуку, або він має обмежені можливості (наприклад, він відповідає тільки на питання «Де?», «Коли?»), тому відповіді на питання користувачів, задані природною мовою, часто опиняються в глибині результатів пошуку. Метапошукова система робить запити відразу кільком пошуковим системам, отримує сніпети з кожної з них (фрагменти тексту, які відображаються поруч з посиланням в результатах пошуку) з топу відповідей певної глибини, об'єднує всі сніпети в єдину видачу і ранжує її так, щоб безпосередні відповіді на питання,

посилання на ревалентні документи та іншу найбільш актуальну інформацію були на початку подачі результатів пошуку.

Далі слід розглянути метод ранжування сніпетів для запитально-відповідального пошуку в метапошуковій системі, в якій використовується інформація, отримана в результаті реляційно-ситуаційного аналізу, що включає семантико–синтаксичний аналіз, а також побудову семантичної мережі. Метод дозволяє знайти відповіді на категориальні питання (тобто питання, в яких на початку зустрічаються питальні слова: «хто», «що», «де», «коли» і т.д.), задані природною мовою.

Метод ранжування сніпетів для питально-відповідального пошуку в метапошуковій системі. Для ранжування сніпетів, отриманих від різних пошукових систем, для кожного з них обчислюється оцінка ревалентності r^{f_j} сніпета $f_j(\text{fragment})$ питальному запиту. В ході обчислення цієї оцінки порівнюється лексика, семантичні ролі, семантичні відношення запитів і сніпета.

Розглянемо загальний алгоритм визначення оцінки ревалентності сніпетів запиту.

Передбачається, що запит складається з одного речення, оскільки запити користувачів до пошукової системи у веб-пошуках, як правило, є короткими. Алгоритм можна легко узагальнити на випадок, коли запит складається з декількох речень.

Сніпет ділиться на речення і для кожного речення обчислюється його оцінка ревалентності до запитального речення. Ця оцінка враховує як лексику, так і семантику речень (семантичні ролі і відношення), тому назвемо цю оцінку лексико-семантичною. Поряд з лексико–семантичною оцінкою розраховується лексична оцінка речення, яка враховує тільки збіг лексики речень сніпета і запиту. Вона необхідна в подальшому для оцінки ревалентності всього сніпета запиту за лексикою. Потрібно врахувати, що при порівнянні лексики стоп–слова, а також питальні слова запиту, такі як «що», «де» і т.д. не враховуються. Крім того, визначається оцінка відповідності фрагмента словникового запасу запиту для всіх речень – лексична оцінка сніпета (f_l^f). Розраховується як відношення потужності

перетину множини лем слів сніпету(L_f)і множини лем слів запиту(L_q) до потужності множини лем слів запиту:

$$r_l^f = \frac{|L_f \cap L_q|}{|L_q|}.$$

Підсумкова оцінка лексико-семантичного фрагменту - це лінійне згортання максимальної оцінки лексико-семантичного речення сніпету та лексичної оцінки сніпета:

$$r_{ls}^f = \alpha_{ls} * \max_{s_i}(r_{ls}^{s_i}) + \alpha_l * r_l^f.$$

Додавання r_l^f до формули ревалентності покликане підвищити ревалентність тих сніпетів, в яких більше слів з запиту, незалежно від того, як вони розподіляються по реченнями, при інших рівних оцінках. На практиці коефіцієнт в лексико-семантичній оцінці повинен бути значно більше коефіцієнта для лексичної оцінки $\alpha_{ls} > \alpha_l$. В експериментах використовувались емпірично підібрані коефіцієнти $\alpha_{ls} = 0,9; \alpha_l = 0,1$.

Для кожного сніпета також обчислюється повна лексична оцінка r_{fl}^f , що представляє суму всіх лексичних оцінок речень сніпета:

$$r_{fl}^f = \sum_i r_l^{s_i}.$$

Після того, як розраховуються повні лексичні оцінки і лексико-семантичні оцінки всіх сніпетів, розраховується остаточно оцінка актуальності кожного сніпета r^f . Це лінійна згортка лексико-семантичного сніпета і повної лексичної оцінки сніпета нормовану на до максимального повну оцінку серед усіх сніпетів:

$$r^f = \beta_{ls} * r_{ls}^f + \beta_l * \frac{r_l^f}{\max_j r_l^{fj}}.$$

На практиці нормована повна лексична оцінка сніпету повинна мало впливати на ревалентність сніпета для питального запиту. Вона додається до формули релевантності, щоб при інших рівних оцінках збільшити ревалентність

сніпетів з великою кількістю лексики, яка співпадає з лексикою запиту і більшим розміром. Більші сніпети, з більшою вірогідністю, містять відповідь, ніж короткі сніпети, що може допомогти при пошуку, якщо оцінки, які спираються на семантику, малі. Крім того, великі сніпети, як правило, містять більш детальну відповідь.

3.3. Застосування семантичного та семантико-синтаксичного аналізу для вирішення проблеми автоматичного вилучення визначень та термінів автора з текстів наукових публікацій

У наукових публікаціях автори часто вводять нові терміни або, для зручності сприйняття статті, дають визначення відомим термінам. Розглянемо завдання вилучення визначень і визначених термінів з текстів наукових публікацій, такі терміни в подальшому будемо називати авторськими термінами..

У прикладних задачах пошуку інформації, визначення і терміни автора можна використовувати для створення термінологічної бази предметної області, поповнення тезаурусів, формування списку ключових слів документа, його анотації або, наприклад, врахування їх особливим чином у формулі розрахунку актуальності при пошуку інформації. Повнотекстовий пошук за визначеннями і термінами автора може допомогти користувачеві зрозуміти значення невідомих термінів при пошуку в збірниках наукових публікацій [27].

Розглянемо метод автоматичного вилучення визначень та термінів автора з текстів наукових публікацій, з використанням морфології, синтаксису та семантики, а також опишемо його експериментальні дослідження на вручну позначеному тілі наукових публікацій.

Спосіб вилучення визначень та термінів автора з текстів наукових публікацій

При вилученні з наукових публікацій термінів автора слід враховувати, що їх тексти можуть містити визначення не тільки вже встановлених, але і нових, раніше невідомих термінів, самі терміни можуть містити іноземні слова і фрази, а публікації можуть бути з різних наукових областей. Все це вимагає гнучкості від

методу вилучення визначень і авторських термінів, наприклад, підхід, заснований на тезаурусах, в даному випадку буде неефективним.

Для вирішення цієї задачі запропоновано метод, заснований на порівнянні лексико-синтаксичної і семантичної структури речення зі списком фреймів (шаблонів). Було запропоновано, крім лексичної, морфологічної та синтаксичної інформації, використовувати семантичні ролі при побудові фреймів. Використання інформації про семантичні ролі дозволяє, з одного боку, спростити створення фреймів, а з іншого боку, охопити більше випадків визначення термінів одним фреймом.

Фахівці проаналізували корпус наукових публікацій, та визначили понад 60 стабільних контекстів, наявність яких свідчить про присутність у реченні визначення терміна. На основі експериментальних даних уточнено контексти і сформовано набір фреймів (всього близько 15), які дозволяють знайти речення, в якому є визначення терміна і виділити підрядок, що відповідає авторському терміну.

Аналізатор вводить структуроване представлення тексту, в якому виділяються слова і речення, вказуються морфологічні характеристики слів, в тому числі леми, виділяються клаузи в реченнях, будуються синтаксичні дерева залежностей і рольові структури речень. Кожне речення тексту зіставлено з фреймами в заданому списку, кожен з яких незалежно від інших вилучає один або кілька термінів. Вилучені терміни потім фільтруються набором евристик. В результаті, в кінці подається список визначених термінів і речень, в яких зустрічаються ці терміни. Речення, які містять вилучені терміни, позначаються як визначення.

Процедура зіставлення фрейму з реченням тексту складається з двох кроків. На першому етапі речення шукає «ключові слова» фреймів. У кожному фреймі є перевірка на наявність певної леми в реченні. Слово, яке має цю лему, буде називатися «ключовим» для фрейму. Перед перевіркою інших умов фрейму шукається текстове речення, яке визначає розташування всіх ключових слів і генерує список фреймів, для яких співпали ключові слова. Такий підхід зручний

для подальшого аналізу, оскільки дозволяє швидко відсіювати речення, які не підпадають під жодний фрейм, і сформувати скорочений список фреймів, які будуть порівнюватися з заданим реченням. На наступному кроці перевіряються інші умови фрейму, які відповідають ключовим словам, знайденим у реченні. Речення в наукових текстах, як правило, мають досить складну структуру, тому зручно перевіряти не всю пропозицію, а в межах однієї клаузи. Наприклад, у реченні «перигелій є точкою орбіти небесного тіла, де воно знаходиться найближче до Сонця» частина «перигелій називається точкою орбіти небесного тіла» спочатку буде проаналізована окремо, а потім частина «де вона найближча до Сонця» і тільки перша частина буде позначена як відповідна кадру 2.

Після знаходження речення, яке задовольняє певний фрейм, безпосередньо виділяється авторський термін. Щоб отримати його, кожен фрейм має бути пов'язаний з ним правилом, яке визначає, яку частину підвиразу слід виділити як термін. Це правило враховує, що термін може складатися з декількох слів.

Більшість термінів знаходиться за допомогою описаного методу пошуку фреймів, однак існує ряд поширених випадків, які викликають помилки при вилученні авторського терміну:

- вибраний вираз містить слова, які не є термінами;
- вирази з запереченням, що містять частинку «ні»;
- вирази, які задають відношення синонімії

Щоб врахувати подібні випадки, після знаходження фрейма і виділення терміну проводиться додаткова процедура фільтрації за допомогою набору евристик.

Для першого випадку існують правила, які фільтрують список слів, включених у вибраний термін. Наприклад, відфільтровуються деякі частки і сполучники («нехай», «якщо», «також» і т.д.), прислівники (традиційно, зазвичай і т.д.).

Щоб відфільтрувати вирази, які підпадають під другий випадок, у відповідній клаузі перевіряється, чи є у ключового слова (якщо воно дієслово) синтаксичне з'єднання з частинкою «не». Частково вдається вирішити проблему з третього

випадку шляхом фільтрації термінів за ключовими словами і формами, в яких ці слова можуть знаходитись.

Експериментальне дослідження методу вилучення визначень та термінів автора з текстів наукових публікацій

Для експериментальної перевірки розробленого методу було відзначено корпус наукових публікацій, що містять журнальні статті зі списку Вищої атестаційної комісії та доповіді на українських наукових конференціях. Розмір зазначеного відсіку становить понад 72 000 токенів, у справі виділено понад 300 визначень. Були розраховані не суворі метрики: точність p , повнота r і F_1 міра. У не строгих метриках правильною відповіддю вважається перетин текстового відрізка в тестовому випадку з відрізком, отриманим автоматично з аналізатора, навіть без точної відповідності країв відрізків. Порівняли результати трьох аналізаторів:

– аналізатор, в якому реалізовані фрейми, які використовують семантичні ролі, отримані з системи семантико0синтаксичного аналізу. Позначимо цей аналізатор як «Sem».

– аналізатор, в якому реалізовані фрейми, які використовують семантичні ролі, отримані від системи семантико-синтаксичного аналізу. Позначимо цей аналізатор як «Sem. – Syn».

– аналізатор, який не має фреймів, які використовують семантичні ролі. Позначимо цей аналізатор як «Sem. – Not». Результати експериментів представлені в таблиці 3.2

Таблиця 3.2 – Оцінки якості методу вилучення авторських термінів з текстів наукових публікацій

| Аналізатор | p , % | r , % | F_1 , % |
|------------|---------|---------|-----------|
| Sem | 82,6 | 68,3 | 74,8 |
| Sem.-Syn. | 82,7 | 68,7 | 75,0 |
| Sem.-Not | 81,6 | 58,3 | 68,0 |

Отримані результати показують, що фрейми з використанням семантичних ролей обробляють значну кількість випадків визначення термінів у наукових публікаціях (більше 10% в повноті), тим самим роблячи значний внесок у вирішення поставленого завдання. Використання семантичної системи розбору не призвело до значного підвищення якості в порівнянні з використанням системи, в якій синтаксичний і семантичний аналіз виконуються окремо. Причина цього в тому, що найбільшу роль у вирішенні цієї задачі відіграють лексикоморфологічні правила, які не враховують семантику і синтаксис речень, тому збільшення повноти визначення рольових структур не сильно впливає на кінцевий результат.

Аналіз помилок показав, що значна кількість термінів, які не були вилучені, вимагають для їх знаходження розбиття кореферентних зв'язків як на рівні одного речення так і на рівні всього тексту. Ще одна значна частина не знайдених термінів, розмічених в перевірочному корпусі, відповідає окремим не частим патернам. На точність негативно впливають помилки фрейму, де визначений термін і його визначення розділені тире або дефісом. Фрейм, який би відповідав лише визначенням в таких випадках, досить важко розрізнити через велику кількість різних лінгвістичних конструкцій з використанням тире або дефісів, а також помилок у їх використанні в текстах природньої мови.

В цілому отримані результати ілюструють застосовність розробленого аналізатора, який реалізує правила, що використовують семантичні ролі, отримані в результаті семантико-синтаксичного аналізу для вирішення задачі вилучення визначень і авторських термінів в прикладних додатках.

Модуль для вилучення визначень і термінів автора використовується в функції повнотекстового пошуку за визначеннями в інформаційно-аналітичній системі Exactus Expert. Вилучені визначення і терміни помічаються в пошуковому індексі окремими тегами. Пошукова система дозволяє вказати обмеження зони пошуку такими тегами. Завдяки цьому користувачі можуть шукати не за повними текстами статей, а за термінами, для яких є означення. Це дозволяє користувачеві швидко знайти визначення невідомого терміну, що дуже корисно в експлоративному пошуку.

3.5 Аналіз та вибір технологій і методів реалізації проектованої системи

Отриманий в попередньому розділі семантико-синтаксичний аналізатор можна застосовувати для чорнової розмітки корпусів тексту. Для створення корпусу тексту необхідна безпосередня участь людини, що робить створення корпусу ресурсовитратним. Семантико-синтаксичний аналізатор допоможе визначити такі основні частини речення, як об'єкт, який виконує дію, тобто підмет, саму дію, тобто присудок та суб'єкт, над яким виробляється дія. Це основні характеристики в реченні, визначення яких автоматично призведе до зниження витрат часу. Однак можуть знадобитися і інші характеристики в реченні, які необхідно визначати людині. До того ж для правильного навчання систем потрібні правильно розмічені корпуси, тому необхідна людська перевірка результатів роботи. Іншим можливим способом використання отриманої системи є перевірка узгодження частин мови між собою. Речення повинне повністю розбиватись на словосполучення. Якщо цього не відбувається, то в реченні є помилка. Помилку в узгодженні слів можна визначити по тому, що граматики знаходить ланцюжок слів, який потенційно може бути словосполученням, але морфологічні правила не пропускають цей ланцюжок як словосполучення, і після цього деякі слова в реченні не відносяться ні до якого словосполучення. Використовуючи граматику і морфологічні правила можна припустити які морфологічні характеристики повинно мати слово без помилки. Такого роду помилки часто виникають при перекладах між мовами зі схожою структурою граматики, але різними морфологічними характеристиками в деяких словах, наприклад, при перекладах між українською та російською мовою, в яких роди багатьох слів відрізняються. Наприклад, слово мова в українській мові має жіночий рід, а в російській – чоловічий.

4.3 Висновки

Продемонстровано застосування розроблених методів семантичного та семантично-синтаксичного аналізу для вирішення прикладних задач обробки тексту природною мовою.

Показано, як інформація про семантичні ролі, отримана від семантичних і семантико-синтаксичних аналізаторів, використовується в алгоритмі побудови семантичної мережі для реляційно-ситуаційної моделі тексту.

Розроблено метод ранжування фрагментів для пошуку питань і відповідей в метапошуковій системі, в якому поряд зі словниковим запасом враховуються семантичні ролі і взаємозв'язки. Експериментально показано, що семантична інформація робить великий внесок у точність алгоритму ранжування і дозволяє вилучати із сніпетів безпосередньо відповіді на питання. Крім того, результати експериментальних досліджень свідчать про значну перевагу використання семантико-синтаксичної системи аналізу при вирішенні задачі пошуку питань-відповідей в порівнянні з системою, в якій синтаксичним і семантичним аналіз виконуються окремо.

Розроблено метод вилучення визначень та термінів автора з текстів наукових публікацій, в якому реалізуються правила, що враховують семантичні ролі. Експериментально показано ефективність розробленого методу і значення внеску семантичних правил у вирішення цієї задачі. Використання рольової структури речення спрощує побудову правил для вилучення визначень та авторських термінів.

Реалізовані методи семантичного і семантико-синтаксичного аналізу, метод побудови семантичної мережі постачання, метод ранжирування фрагментів для пошуку запитань-відповідей, а також метод вилучення визначень і визначених термінів в пошуково-пошуково-аналітичних системах Exactus, Exactus Expert.

4 РЕАЛІЗАЦІЯ МЕТОДІВ СЕМАНТИЧНОГО ТА СЕМАНТИКО-СИНТАКСИЧНОГО АНАЛІЗУ

4.1 Використання семантичного та семантико-синтаксичного аналізу для побудови семантичної мережі реляційно-ситуаційної моделі

Тестування розроблених методів семантичного і семантико-синтаксичного аналізу проводилося на корпусі ГРАК. Генеральний регіонально анотований корпус української мови (ГРАК; англ. General Regionally Annotated Corpus of Ukrainian, GRAC) — це корпус української мови обсягом понад 800 млн токенів, призначений для здійснення лінгвістичних досліджень з граматики, лексики, історії української літературної мови, а також для використання під час укладання словників та граматик.

Корпус можна використати для вивчення мови, також при підготовці навчальних матеріалів, підручників, навчальних словників та вправ з використанням прикладів з реальних текстів, з урахуванням частотної сполучуваності тощо. Корпус не є взірцем нормативної української мови, в ньому можуть трапитися слова і сполучення, які не відповідають сучасним нормам літературної мови. Наведені далі статистичні дані стосуються 10 версії корпусу, доступної для пошуку з 20 жовтня 2020 року.

Корпус охоплює період з 1816 по 2021 р. і містить понад 80 тисяч текстів близько 23000 авторів.

Для оцінки якості вирішення задачі визначення рольових структур тверджень створено семантичний корпус. Він розмічує предикатні слова, визначає їх семантичні аргументи і показує їх семантичні ролі. Підкорпус також містить категорично-семантичні класи іменників, для яких вказується семантична роль. Відмітимо, що розмічені предикатні слова і ролі тільки для тих випадків, які описані семантичним словником.

При тестуванні багатьох методів використовують наступні метрики: точність p , повнота r і $F1$ -міра. Вони розраховуються наступним чином:

$$p = \frac{tp}{tp + fp}; r = \frac{tp}{tp + fn}; F_1 = \frac{2 * p * r}{(p + r)}.$$

Загалом, tp (true positives) - це кількість правильно розпізнаних позитивних прикладів; fp (false positives) – кількість прикладів, які помилково віднесені до позитивних; fn (false negatives) - це кількість позитивних прикладів, які були помилково класифіковані як негативні. У кожному випадку обумовлюється, що мається на увазі під tp , fp , fn .

4.2. Експериментальне дослідження методу пошуку семантичних аргументів

Метод пошуку семантичних аргументів тестувався при наступних умовах (конфігурацій):

- 1) морфологічні особливості та синтаксична розмітка від золотого стандарту подаються на вході аналізатора. Позначаємо таку конфігурацію, як «Gold_Morph_Synt».
- 2) на вхід аналізатора подаються морфологічні признаки і синтаксична розмітка, що генеруються автоматично. Позначаємо таку конфігурацію, як «Morph_Synt». Ця конфігурація відображає реальну завдачу аналізу тексту в деякій прикладній системі обробки текстів на природній мові

Для тестування методу пошуку семантичних аргументів було використано розмічений корпус з 200 речень, в якому для кожного предикатного слова були розмічені всі його семантичні аргументи. Розраховувались точність p , повнота r і $F1$ міра. У цих оцінках:

tp - кількість семантичних аргументів у виході аналізатора, що співпали з семантичними аргументами в тестовому корпусі, у яких також співпали позиції предикатних слів у реченні;

fp – кількість семантичних аргументів у видачі аналізатора, які відсутні в тестовому корпусі, або у них відрізняються позиції предикатних слів;

fn – кількість семантичних аргументів у тестовому випадку, які відсутні на видачі аналізатора, або вони не співпадають позиції предикатних слів

Результати наведені в таблиці 4.1.

Таблиця 4.1 – Результати експериментальної оцінки якості пошуку семантичних аргументів

| Конфігурація | $p, \%$ | $r, \%$ | $F_1, \%$ |
|-----------------|---------|---------|-----------|
| Gold_Morph_Synt | 94,4 | 95,0 | 94,6 |
| Morph_Synt | 86,1 | 83,2 | 84,6 |

Результати показують, що розроблений евристичний метод пошуку семантичних аргументів з ідеальною синтаксичною розміткою, як і очікувалося, працює з високою точністю і повнотою. Це дозволяє використовувати його для пошуку основних семантичних аргументів в системі семантико-синтаксичного аналізу. Помилки найчастіше виникають, якщо предикатне слово і його аргументи пов'язані через досить складні синтаксичні конструкції, наприклад, у випадках, коли в одній клаузі є кілька однорідних предикатних слів. Якість пошуку семантичних аргументів значно знижується при використанні автоматично згенерованої розмітки синтаксису – приблизно на 10% відповідно по F_1 -міри. Експеримент показує важливість якісного синтаксичного аналізу як для розв'язку задачі пошуку семантичних аргументів, так і для всієї процедури даного семантичного аналізу.

4.2. Результати експериментів методу семантичного аналізу

Семантичний аналізатор був протестований за наступних умов (конфігурацій):

1) морфологічні особливості, синтаксична розмітка і КСК від золотого стандарту подаються на вхід аналізатора. Позначаємо таку конфігурацію, як «Gold_Morph_CSC_Synt».

2) морфологічні особливості та синтаксичні відмітини від золотого стандарту подаються на вхід аналізатора, а КСК визначаються автоматично. Позначаємо таку конфігурацію, як «CSC_Gold_Morph_Synt»

3) на вхід аналізатора подається морфологічні ознаки КСК від золотого стандарту, а синтаксична розмітка генерується автоматично за допомогою синтаксичного аналізатора. Позначаємо таку конфігурацію, як «Synt_Gold_Morph_CSC».

4) морфологічні ознаки від золотого стандарту подаються на вхід аналізатора, а КСК і синтаксична розмітка генеруються автоматично аналізаторами. Ця конфігурація використовує модель синтаксичного аналізатора, навчену морфологічними особливостями із золотого стандарту. Позначаємо таку конфігурацію, як «CSC_Synt_Gold_Morph».

5) всі знаки і вся розмітка, що подаються на вхід семантичного аналізатора, генеруються автоматично. Ця конфігурація відповідає реальному практичному завданню аналізу тексту: просто текст подається на вхід аналізатора, а на виході будується дерево синтаксису і рольова структура. Позначаємо таку конфігурацію, як «Morph_CSC_Synt».

Для тестування аналізатора для визначення рольових структур тверджень було прийнято близько 1500 пропозицій, що становить приблизно 25 000 жетонів без урахування пунктуації. У тестовому підкорпусі розмічено близько 3300 ролей. Інша частина семантично розміченого корпусу була використана для налаштування аналізатора.

Розраховувались точність p , повнота r і F_1 -міра. При порівнянні результатів аналізатора з тестовим підкорпусом враховувалися тільки ті семантичні ролі, які присутні в тестовому підкорпусі. В цьому випадку:

tp – кількість аргументів, роль яких при заданому предикатному слові співпадає з роллю, вказаною в тестовому підкорпусі при цьому предикатному слові

fn – кількість аргументів, роль яких при заданому предикатному слові вказана в текстовому підкорпусі, але в аналізаторі на виході роль для цього аргументу при заданому предикатному слові не визначена

fp – кількість аргументів, роль яких при заданому предикатному слові вказана в тестовому підкорпусі, у видачі аналізатора роль для цього аргументу при заданому предикатному слові визначена але не співпадає з роллю, вказаною в підкорпусі.

Результати оцінки якості вирішення задачі визначення рольових структур тверджень представлені в таблиці 4.2.

Таблиця 4.2 – Результати експериментальної оцінки якості семантичного аналізатора.

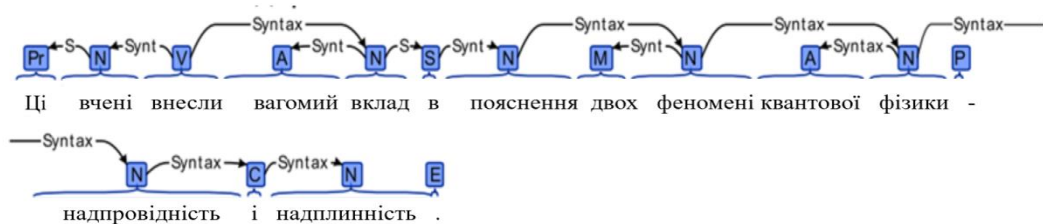
| Конфігурація | p , % | r , % | F_1 , % |
|---------------------|---------|---------|-----------|
| Gold_Morph_CSC_Synt | 92,3 | 80,8 | 86,2 |
| CSC_Gold_Morph_Synt | 89,4 | 73,4 | 80,6 |
| Synt_Gold_Morph_CSC | 92,1 | 74,3 | 82,3 |
| CSC_Synt_Gold_Morph | 89,2 | 67,3 | 76,7 |
| Morph_CSC_Synt | 89,6 | 61,0 | 72,6 |

Отримані результати свідчать про те, що розроблений семантичний аналізатор для аналізу текстів українською мовою демонструє якість вирішення проблеми визначення рольових структур тверджень на рівні сучасних аналізаторів для інших мов з невеликою кількістю мовних ресурсів. Виходячи з отриманих результатів, можна судити про важливість різних признаков для вирішення цієї задачі. Перш за все, необхідно зазначити великий вплив на кінцевий результат

помилки при визначенні категоріально-семантичних класів. При переході від конфігурації «Gold_Morph_CSC_Synt» до «CSC_Gold_Morph_Synt» значно знижується як точність (близько 3%), так і повнота (близько 7,5%). Для підвищення точності визначення слів КСК можна використовувати методи кластеризації великих колекцій, а також лінгвістичних тезаурусів. Використання автоматично згенерованої розмітки в основному призводить лише до зниження повноти: при переході від конфігурації «Gold_Morph_CSC_Synt» до конфігурації «Synt_Gold_Morph_CSC» повнота знижується приблизно на 6,5%. Це є відображення помилок, які виникають в процесі семантичних аргументів. При переході від конфігурацій, в яких морфологічні признаки беруться від золотого стандарту, до конфігурації, в якій морфологічні признаки генеруються автоматично («Morph_CSC_Synt»), накладаються як помилки в порівнянні морфологічних особливостей ролей і аргументів, так і зниження точності синтаксичного аналізу.

Як видно з результатів, лише близько половини всіх помилок у визначенні рольових структур висловлень пов'язані з помилками на попередніх етапах обробки тексту: морфологічному, синтаксичному аналізу і визначення КСК. В іншій половині значні ролі відіграють помилки безпосередньо в методах визначення рольових структур тверджень. Ще однією проблемою семантичного аналізатора, яка викликає значну частку помилок, є обмеження семантичного словника. Словникові записи семантичного словника містять лише три особливості диференціювання семантичних ролей, яких іноді буває недостатньо. Існує значна кількість випадків, коли ці ознаки не можуть бути використані для розділення двох семантичних ролей. Частина цієї проблеми вирішується евристичними методами. Ще одним обмеженням семантичного словника є помилки, що містять рисунки 4.1 і 4.2, показують приклади семантичного аналізу речень від корпусу ГРАК. Зелений вказує на присудки слів, червоний вказує на аргументи. Червоні стрілки вказують на семантичні ролі, зелені стрілки вказують на семантичні відносини в семантичній мережі.

Синтаксичне дерево

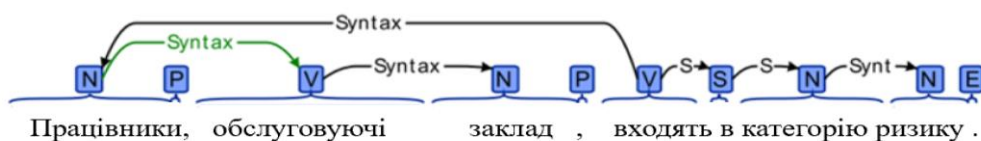


Рольова структура речення



Рисунок 4.1 – Результат семантичного аналізу речення «Ці вчені зробили вирішальний внесок у пояснення двох явищ квантової фізики – надпровідності і надплинності».

Синтаксичне дерево



Рольова структура речення

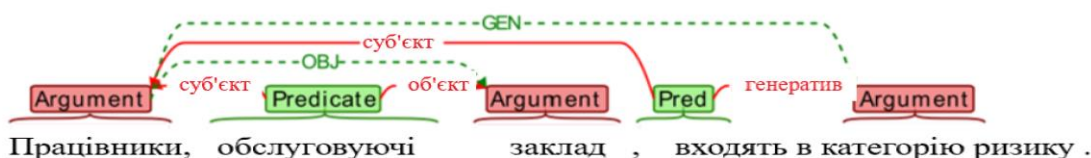


Рисунок 4.2 – Результат семантичного аналізу речення «Працівники, які обслуговують заклад, знаходяться в категорії ризику».

4.4. Результати експериментів методу семантико-синтаксичного аналізу.

Оцінка впливу методу семантико-синтаксичного аналізу на якість синтаксичного аналізу.

Для того щоб оцінити вплив розробленого методу на якість побудови синтаксичних дерев залежностей, система семантико-синтаксичного аналізу порівнювалась з синтаксичним аналізатором, реалізованим за допомогою MaltParser. Він також використовувався для попереднього синтаксичного аналізу в системі семантико-синтаксичного аналізу. Тестування і навчання здійснювалися на корпусі ГРАК

На текстовому підкорпусі проводилась оцінка якості встановлення синтаксичних зв'язків між предикатним словом і семантичними аргументами. Враховувалися тільки синтаксичні зв'язки, де основним словом є дієслово, дієприкметник, а залежним – є іменник, займенник, прийменник, якому підпорядкований іменник або займенник, питальне слово, а також зв'язки, де головне - іменник або займенник, а залежний - дієприкметник. В якості оціночних метрик розраховували точність p , повнота r і F_1 -міра. Тут:

– tp – кількість синтаксичних зв'язків між предикатним словом та їх аргументами в тестовому підкорпусі, які співпали з зв'язками при видачі аналізатора;

– fp – кількість зв'язків поданій аналізатором, встановлених між предикатним словом і словами, які мають ознаки аргументу, які відсутні в тестовому підкорпусі;

– fn – кількість зв'язків між предикатним словом та їх аргументами, які присутні в тестовому підкорпусі, але відсутні у видачі аналізатора.

В експериментах порівнювали два способи перевірки виправлень в семантико-синтаксичному аналізі: за допомогою використання статистично-евристичного критерію, заснованого на перевірці частоти спільного виникнення ознак слів, і критерій, заснований на ансамблі бінарних класифікаторів за допомогою машинного навчання. Таким чином, порівнювалися три аналізатори:

- синтаксичний аналізатор на основі MaltParser, назвемо його «MaltParser»;

– система семантико–синтаксичного аналізу, в якій реалізований статистично-евристичний критерій перевірки виправлень, позначаємо її як «Sem. –sin. Stat.»;

– система семантико–синтаксичного аналізу, яка реалізує критерій перевірки виправлень за допомогою класифікаторів на основі машинного навчання, позначимо її як «Sem. –sin. MO».

Результати експериментальних досліджень цих аналізаторів на текстовому синтаксично розміченому корпусі представлені в таблиці 4.3.

Таблиця 4.3 – Порівняння якості синтаксичного аналізу системи аналізу з аналізатором MaltParser

| Аналізатор | p , % | r , % | F_1 , % |
|-----------------|---------|---------|-----------|
| MaltParser | 90,6 | 89,0 | 89,8 |
| Sem.-sin. Stat. | 90,5 | 90,2 | 90,3 |
| Sem.-sin. MO | 90,6 | 90,1 | 90,4 |

Отримані результати показують, що розроблений метод семантико-синтаксичного аналізу дозволяє підвищити повноту встановлення синтаксичних зв'язків між предикатними словами і аргументами більш ніж на 1,1%, тим самим йому вдається встановити більше 10% всіх найскладніших синтаксичних зв'язків між предикатними словами і аргументами, які не були знайдені MaltParser. При цьому точність системи семантико-синтаксичного аналізу не змінюється. Відсутність значного поліпшення точності пов'язано з тим, що в отриманих оцінках враховують тільки нові зв'язки, додані до синтаксичного дерева, але не враховують віддалені. Поліпшення повноти встановлення синтаксичних зв'язків має значний вплив на збільшення міри F_1 .

Статистична значимість отриманих результатів була перевірена. Для цього був використаний метод статистичного бутстрепінга. Показано, що приріст повноти і міри системи в порівнянні з MaltParser є статистично значущим. Довірчий інтервал на рівні значущості $\alpha = 0,05$ для повноти був (0,9; 1,2)%.

Довірчий інтервал для міри на рівні значущості $\alpha = 0,05$ був $(0,4; 0,6)\%$.

Експериментальне порівняння методів корекції фільтрації показало, що в цілому метод, заснований на ансамблі бінарних класифікаторів, працює краще, ніж статистично-евристичний критерій. Він показує більшу точність при невеликому зниженні повноти. Однак варто відзначити, що досить простий критерій також дозволяє досить добре відфільтрувати неправильні виправлення. Тестування показало, що точність внесення виправлень до дерева синтаксису за допомогою семантико-синтаксичної системи становить 82-85%, в залежності від методу перевірки виправлень, параметрів класифікаторів, а також порогу вирішального правила. Рисунки 4.3 і 4.4 наводять приклади, в яких система семантико-синтаксичного аналізу виправляє синтаксичні структури речень.

А) Синтаксичне дерево

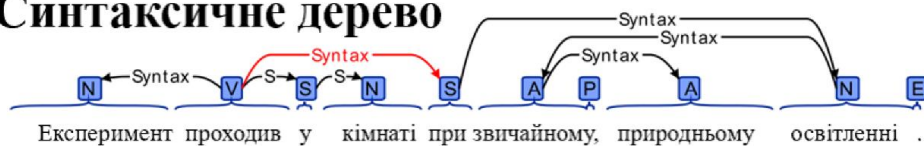


Рольова структура речення

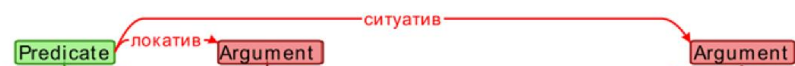


Експеримент проходив у кімнаті при звичайному, природньому освітленні

Б) Синтаксичне дерево



Рольова структура речення

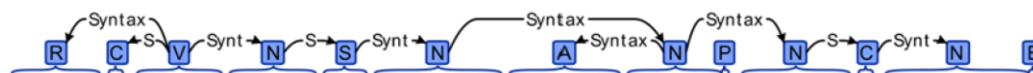


Експеримент проходив у кімнаті при звичайному, природньому освітленні

Рисунок 4.3 – Приклад, в якому семантична система розбору виправила синтаксичне з'єднання. На рисунку А) – приклад роботи MaltParser, на рисунку Б) – приклад роботи системи семантико-синтаксичного аналізу

У цих малюнках синтаксичні зв'язки, які були додані семантичною системою аналізу, виділяються червоним кольором, в результаті чого він також зміг побудувати більш повні рольові структури.

А) Синтаксичне дерево



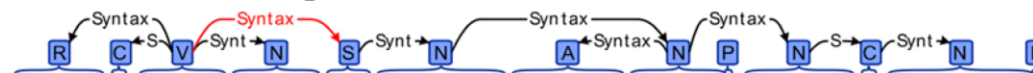
Тому і б'ються вчені над створенням плавучих міст, островів і будинків.

Рольова структура речення



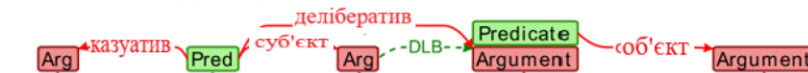
Тому і б'ються вчені над створенням плавучих міст, островів і будинків

Б) Синтаксичне дерево



Тому і б'ються вчені над створенням плавучих міст, островів і будинків.

Рольова структура речення



Тому і б'ються вчені над створенням плавучих міст, островів і будинків

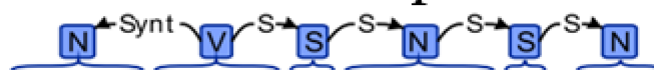
Рисунок 4.4 – Приклад, в якому система семантико-синтаксична аналізу виправила синтаксичний зв'язок. На рисунку А) – приклад MaltParser, на рисунку Б) – приклад семантичної системи семантико-синтаксичного аналізу

Не зважаючи на те, що процедура коригування системи семантико-синтаксичного аналізу працює з високою точністю, в деяких випадках можуть виникати помилки, в результаті чого правильний синтаксичний зв'язок змінюється на неправильний. По-перше, бувають випадки семантичної омонімії, коли на рівні аналізу одного речення неможливо (навіть для носія мови) вирішити неоднозначність побудови як семантичної, так і синтаксичної структури речення. По-друге, існують ситуації, в яких два слова «конфліктують» за право приєднати до себе один і той же семантичний аргумент, щоб заповнити деяку свою семантичну валентність. Якщо семантичний словник аналізатора містить знання про валентності обох слів, то таку ситуацію потенційно не складно вирішити.

Однак у семантичному словнику є інформація про рольові структури не всіх предикатних слів. Хоча охоплення дієслів досить високе, рольові структури багатьох іменників залишаються невідомими. Тому статистичні методи використовувалися для вирішення неоднозначності встановлення синтаксичні зв'язки для цих випадків. Але статистично-евристичний критерій і класифікатори, на основі машинного навчання, не завжди можуть точно вирішити такі ситуації.

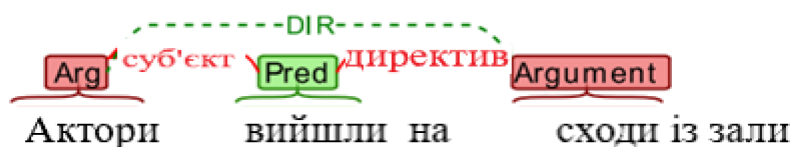
На рисунку 4.5 показаний приклад, в якому система семантико-синтаксичного аналізу правильно коригує дерево синтаксису, а на рисунку 4.6 представлений схожий приклад, в якому коригування дерева призводить до того, що з'являється помилка.

А) Синтаксичне дерево

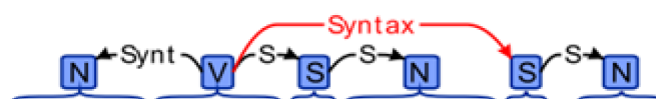


Актери вийшли на сходи із зали

Рольова структура речення



Б) Синтаксичне дерево



Актери вийшли на сходи із зали

Рольова структура речення

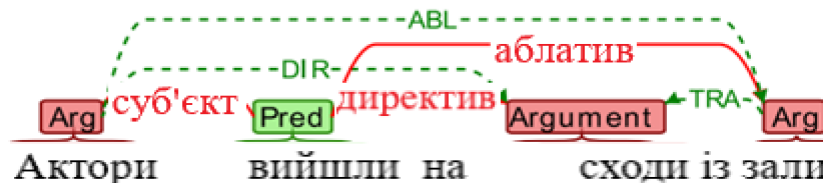
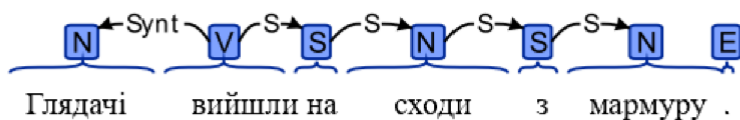
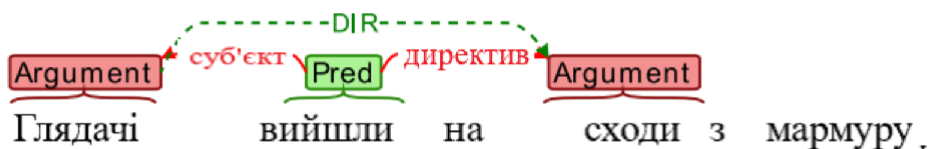


Рисунок 4.5 – Приклад, в якому в результаті коригування дерева синтаксису виправлено помилку

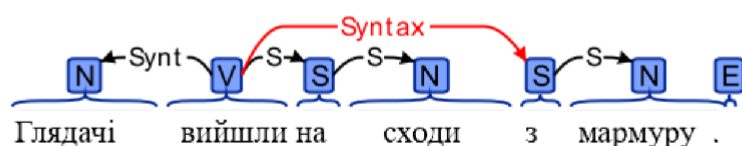
А) Синтаксичне дерево



Рольова структура речення



Б) Синтаксичне дерево



Рольова структура речення

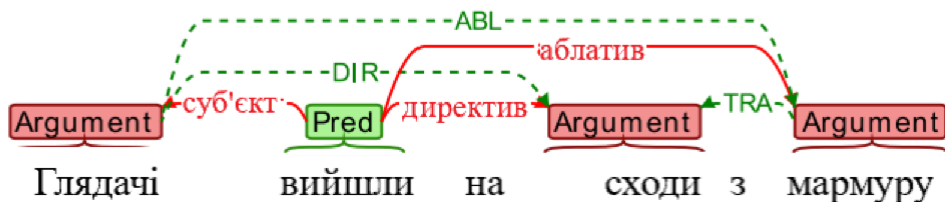


Рисунок 4.6 – Приклад, в якому виправлення дерева синтаксису призвело до помилки

Підвищення якості як синтаксичного аналізу, так і семантичного аналізу за допомогою розробленого методу семантико-синтаксичного аналізу тісно пов'язане з повнотою семантичного словника.

4.5. Оцінка ефективності методу семантико-синтаксичного аналізу на якість вирішення задачі визначення рольових структур тверджень

У проведених експериментах порівнювалась якість вирішення задачі визначення рольових структур тверджень семантико-синтаксичного аналізу, яку позначимо як «Sem. –sin.», і системою, в якій синтаксичний і семантичний аналіз

виконуються окремо, позначається як «Основний». В системі семантико–синтаксичного аналізу був використаний метод перевірки виправлень за допомогою ансамблю бінарних класифікаторів на основі машинного навчання.

Тестування системи семантико-синтаксичного аналізу проводилося аналогічно тестуванню семантичного аналізатора було використано(семантично позначений підкорпус (1500 речень, приблизно 25000 токенів без пунктуації, 3300 ролей), розраховано точність p , повнота r і F_1 -міра. Результати представлені в таблиці 4.4

Таблиця 4.4 – Порівняння якості семантичного аналізу системи семантико-синтаксичного аналізу з системою, в якій синтаксичним і семантичний аналіз виконуються окремо

| Система | p , % | r , % | F_1 , % |
|-----------|---------|---------|-----------|
| Базова | 89,6 | 61,0 | 72,6 |
| Sem.-sin. | 89,6 | 62,7 | 73,8 |

Результати показують, що застосування розробленого методу семантико-синтаксичного аналізу призводить до підвищення якості вирішення задачі визначення рольових структур тверджень. Зокрема, система семантико-синтаксичного аналізу має більшу повноту – на 1,7%, що призводить до збільшення F_1 -міри на 1,2%. При цьому точність не зменшується

Завдяки коригуванню синтаксичної структури збільшується частка правильно ідентифікованих семантичних аргументів, що в кінцевому підсумку дозволяє підвищити якість семантичного аналізу в цілому. На рисунках 27 і 28 показані приклади, в яких семантико–синтаксична система аналізу побудувала більш повні рольові структури, ніж система, в якій синтаксичний і семантичний аналіз виконуються окремо.

4.6 Програмна реалізація

Була реалізована програмна система. Система написана на мові програмування C# з використанням технологій ASP.NET та Entity Framework. В якості програмної середовища використовувалось MS Visual Studio. Програмна система складається з декількох класів: – клас, який виконує морфологічний аналіз. Він повертає морфологічні ознаки слів. Для цього використовується зовнішня бібліотека AOT; – клас, який зберігає морфологічні правила для контекстно-вільної граматики та таблиці action, goto; – клас, який виконує LR-алгоритм; – клас, який зберігає отриманий результат у базу даних;

– клас, який відповідає за зв'язок зовнішнього інтерфейсу з класами, які проводять аналіз даних. Цей клас збирає текст, введений користувачем та відображає результат аналізу. Програма дозволяє користувачу ввести текст, який він хоче проаналізувати. Аналіз можна запустити кнопкою під вікном введення тексту. Після аналізу користувач отримує розбиття кожного речення на словосполучення. В разі коли у реченні була помилка або програма не змогла зробити аналіз, видається повідомлення про це, та вказується на якому етапі трапилась помилка: під час морфологічного аналізу (це значить програма не змогла розпізнати окреме слово, в слові помилка чи цього слова нема у словнику), чи під час синтаксичного аналізу (проблема в узгодженні слів в реченні).

4.5. Висновки

Представлено результати експериментальних досліджень розроблених методів семантичного та семантико-синтаксичного аналізу на маркованих корпусах українськомовних текстів.

Отримано результати оцінки якості вирішення проблеми визначення рольових структур тверджень у текстах українською мовою. Результати показують, що якість вирішення цієї задачі за допомогою розробленого методу відповідає рівню сучасних семантичних аналізаторів для інших мов з невеликою кількістю мовних ресурсів.

Досліджено вплив розробленого методу семантико-синтаксичного аналізу на якість побудови синтаксичною залежністю дерев та на якість вирішення задачі визначення рольових структур тверджень. семантико-синтаксична система аналізу порівнювалася з синтаксичним аналізатором MaltParser, який зарекомендував себе як один з кращих синтаксичних аналізаторів на основі машинного навчання для україномовних текстів. Експерименти показали, що розроблений метод семантико-синтаксичного аналізу дозволяє підвищити повноту встановлення синтаксичних зв'язків. Системі семантико-синтаксична аналізу вдається виправити найскладніші випадки встановлення синтаксичних зв'язків між предикатними словами і аргументами. Незважаючи на те, що початковий синтаксичний аналізатор показує високу повноту – трохи менше 90%, за допомогою системи семантико-синтаксичного аналізу вдається відновити більше 10% відсутніх синтаксичних зв'язків між предикатними словами і аргументами. Таким чином, вперше досліджено вплив інформації, закладеної в рольову структуру тверджень на якість встановлення синтаксичних зв'язків між предикатними словами та їх семантичними аргументами, та показано важливість цієї інформації для коригування дерева синтаксичної залежності.

Результати експериментів показують, що метод семантико-синтаксичного аналізу також дозволяє підвищити повноту і F_1 -міру визначення рольових структур тверджень. Далі важливість використання семантичного аналізу при вирішенні прикладних завдань обробки тексту природною мовою, а також демонструє ефект підвищення якості синтаксичного і семантичного аналізу, досягнутий за рахунок використання розробленого методу семантико-синтаксичного аналізу.

ВИСНОВКИ

Препроцесінг вхідного текстового коду не можливий без використання різного виду аналізів. Одним із провідних її видів є семантичний та семантико-синтаксичний аналіз. В даній роботі було проаналізовано та удосконалено ряд завдань та методів, а саме:

1. Розроблено новий метод автоматичного визначення рольових структур висловлень, заснований на комунікативній граматиці української мови.

2. Розроблено новий метод комп'ютерного семантико-синтаксичного аналізу текстів, в якому інтегровані методи побудови синтаксичною дерев залежностей і визначення рольових структур тверджень, що дозволяє підвищити точність і повноту синтаксичного і семантичного аналізу в порівнянні з реалізацією, в якій ці види аналізу виконуються окремо.

3. Розроблена та впроваджена інтегрована система семантико-синтаксичного аналізу. Система використовується для вирішення задач пошуку питань-відповідей, вилучення визначень і авторських термінів з текстів наукових публікацій.

4. Експериментально показано, що використання рольової структури речення підвищує ефективність побудови правил вилучення визначень та термінів автора з текстів наукових публікацій.

За темою і результатами дипломної роботи опубліковані тези доповіді на науково-практичній -конференції (додаток В) та статті у фаховому та Міжнародному науковому журналі (додаток В).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бевзенко С. П. Структура складного речення в українській мові. – К.: Вид-во Київ. держ. пед. ін-ту, 1987. – 79 с.
2. Горяний В. Д. Односкладні речення в українській мові. – К.: Рад. шк., 1983. – 165 с.
3. Грищенко А. П. Складносурядне речення в сучасній українській літературній мові. – К.: Наук, думка, 1969. – 155 с
4. A Comprehensive Introduction to Word Vector Representations [Електронний ресурс] – Режим доступу: <https://medium.com/ai-society/jkljlj-7d6e699895c4> – 30.05.2018
5. Boutell M. Learning multi-label scene classification. Pattern Recognition [Електронний ресурс] / M. Boutell, J. Luo, X. Shen, C. Brown// Science Direct, 2004. — P. 1757-1771. — Режим доступу: <https://www.rosehulman.edu/~boutell/publications/boutell04PRmultilabel.pdf>
6. Godbole S. Discriminative methods formulti—labeled classification / S. Godbole, S. Sarawagi // Adv.Knowl. Discov. Data Mining, 2004. — P. 22- 30.
7. Herrera F. Multilabel classification: problem analysis, metrics and techniques / F. Herrera, F. Charte, A. J. Rivera, M. J. del Jesus // Springer International Publishing Switzerland, 2016. — P. 65-78.
8. Machine learning text feature extraction (tf-idf) [Електронний ресурс] – Режим доступу: <http://blog.christianperone.com/2011/10/machine-learningtext-feature-extraction-tf-idf-part-ii/> – 30.11.2018
9. Read J. Classifier chains for multi-label classification [Електронний ресурс] / J. Read, B. Pfahringer, G. Holmes, E. Frank // Department of Computer Science, 2011. — P. 5-16. — Режим доступу: https://www.researchgate.net/publication/227319967_Classifier_Chains_for_Multi-label_Classification
10. Taylor J. An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements / J. Taylor. — University Science Books, 1999. — P. 128-129.

11. Word2Vec [Електронний ресурс] – Режим доступу: <https://www.slideshare.net/HyunYoungLee3/word2-vec-81790245> – 30.05.2018
12. Арполенко Г. П., Забеліна В. П. Структурно-семантична будова речення в сучасній українській мові. – К.: Наук, думка, 1982. – 132 с.
13. Вихованець І. Р. Нариси з функціонального синтаксису української мови. – К.: Наук, думка, 1992. – 224 с.
14. Вихованець І. Р., Городенська К. Г., Русанівський В. М. Семантико-синтаксична структура речення. – К.: Наук, думка, 1983. – 219 с.
15. Городенська К. Г. Дери́вація синтаксичних одиниць. – К.: Наук, думка, 1991. – 192 с.
16. Дзех М.Ю. Формализация толкового словаря естественного языка / Д.В.Преснякова, М.Ю.Дзех // Радиоэлектроника и молодежь в XXI веке: 14-й международный молодежный форум: материалы форума. – Харьков: ХНУРЭ. – 2010.–С. 219.
17. Дюбко Г.Ф. Формальная семантика и анализ естественного языка / Г.Ф.Дюбко, Д.В.Преснякова // Горизонти прикладної лінгвістики та лінгвістичних технологій: Міжнародна наукова конференція. – Партеніт. – 2008. – С. 153–154.
18. Дюбко Г.Ф. Модель поверхностного смысла естественного языка на базе семантических функций / Г.Ф.Дюбко, Д.В.Преснякова // Бионика интеллекта. – 2007. – Вып. 1(66). – С. 103–106.
19. І.М. Шпінарева, О.А. Геренко, К.Ю. Морозова Класифікація текстів на природній мові за допомогою нейронної мережі / І.М. Шпінарева, О.А. Геренко, К.Ю. Морозова – Збірник наукових праць Військового інституту Київського національного університету ім. Т.Шевченка. – К.: ВІКНУ, 2018, Вип. №59 – С.178-184.
20. Немного про word2vec: полезная теория [Електронний ресурс] – Режим доступу: <http://nlrx.net/archives/179> – 30.05.2018
21. Омельченко В.И. Синтаксико–семантический анализ в естественном языке / В.И.Омельченко, Д.В.Преснякова // Радиоэлектроника и молодежь в XXI

веке: 14–й международный молодежный форум: материалы форума. – Харьков: ХНУРЭ. – 2010.–С. 220.

22. Преснякова Д.В. Исследование формальной модели смысла естественного языка на базе семантических функций / Д.В.Преснякова // Радиоэлектроника и молодежь в XXI веке: 9–й международный молодежный форум: материалы форума. – Харьков: ХНУРЭ. – 2005. – С. 372

23. Шинкаренко В. И. Конструктивно-продукционная модель графowego представления текста / В. И. Шинкаренко, Е. С. Куропятник – Проблемы программирования // науч. журнал. – № 2 – 3. – 2015. – с. 63 – 72. Передрук: Shynkarenko V. Constructive-synthesizing model of text graph representation // V. Shynkarenko, O. Kuropiatnyk – CEUR Workshop Proceedings. – 2016. – Vol. 1631. – P. 63 – 72.

24. Шинкаренко В. И. Объектно-ориентированная модель смысловых составляющих языковых конструкций / В. И. Шинкаренко, Е. С. Куропятник. – Искусственный интеллект // науч. журнал. – НАН Украины, Институт проблем искусственного интеллекта. – № 4. – 2014. – с. 44 – 56

25. Шинкаренко В. І. Система контролю плагіату в студентських роботах / В. І. Шинкаренко, О. С. Куроп'ятник // Восточно-Европейский журнал передовых технологий //научный журнал. – Харьков: Технологический центр, 2012 – № 4/2 (58). – с. 32 – 36..

ДОДАТОК А
(обов'язковий)

ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ

depslib dependency file v1.0

1443993032 source:d:\project\antiplagiat\main.cpp

<iostream>
<fstream>
<map>
<iterator>
<cmath>
<cstdio>
<boost\regex.hpp>
«cheker.h»

1443976630 d:\project\antiplagiat\cheker.h

<boost\regex.hpp>
<string>

1443596221 d:\libs\include\boost-1_53\boost\regex.hpp

<boost/regex/config.hpp>
<boost/regex/v4/regex.hpp>

1443596336 d:\libs\include\boost-1_53\boost\regex\config.hpp

<boost/regex/config/borland.hpp>
<boost/config.hpp>
<stdlib.h>
<stddef.h>
<wctype.h>
<boost/regex/config/cwchar.hpp>
<string>
<boost/config/auto_link.hpp>
<stdexcept>
<string>
<boost/throw_exception.hpp>

1443596550 d:\libs\include\boost-1_53\boost\regex\config\borland.hpp

<cstring>

1443596215 d:\libs\include\boost-1_53\boost\config.hpp

<boost/config/select_compiler_config.hpp>
<boost/config/select_stdlib_config.hpp>
<boost/config/select_platform_config.hpp>

<boost/config/suffix.hpp>

1443596236 d:\libs\include\boost-1_53\boost\config\select_compiler_config.hpp

1443596236 d:\libs\include\boost-1_53\boost\config\select_stdlib_config.hpp

<cstddef>

<stddef.h>

<boost/config/no_tr1/utility.hpp>

1443596405 d:\libs\include\boost-1_53\boost\config\no_tr1\utility.hpp

<utility>

1443596236 d:\libs\include\boost-1_53\boost\config\select_platform_config.hpp

<boost/config/posix_features.hpp>

1443596236 d:\libs\include\boost-1_53\boost\config\posix_features.hpp

<unistd.h>

1443596236 d:\libs\include\boost-1_53\boost\config\suffix.hpp

<limits.h>

<cstddef>

«boost/type.hpp»

«boost/non_type.hpp»

<typeinfo>

1443596222 d:\libs\include\boost-1_53\boost\type.hpp

1443596219 d:\libs\include\boost-1_53\boost\non_type.hpp

1443596550 d:\libs\include\boost-1_53\boost\regex\config\cwchar.hpp

<cwchar>

<cwctype>

<boost/config.hpp>

1443596236 d:\libs\include\boost-1_53\boost\config\auto_link.hpp

<boost/config.hpp>

<boost/version.hpp>

1443596223 d:\libs\include\boost-1_53\boost\version.hpp

1443596222 d:\libs\include\boost-1_53\boost\throw_exception.hpp

<boost/exception/detail/attribute_noreturn.hpp>

<boost/detail/workaround.hpp>

<boost/config.hpp>

<exception>

<boost/exception/exception.hpp>

<boost/current_function.hpp>

1443596415 d:\libs\include\boost-1_53\boost\exception\detail\attribute_noreturn.hpp

1443596246 d:\libs\include\boost-1_53\boost\detail\workaround.hpp
<boost/config.hpp>

1443596247 d:\libs\include\boost-1_53\boost\exception\exception.hpp

1443596216 d:\libs\include\boost-1_53\boost\current_function.hpp

1443596552 d:\libs\include\boost-1_53\boost\regex\v4\regex.hpp

<boost/regex/config.hpp>
<boost/regex/v4/regex_workaround.hpp>
<boost/regex_fwd.hpp>
<boost/regex/regex_traits.hpp>
<boost/regex/v4/error_type.hpp>
<boost/regex/v4/match_flags.hpp>
<boost/regex/v4/regex_raw_buffer.hpp>
<boost/regex/pattern_except.hpp>
<boost/regex/v4/char_regex_traits.hpp>
<boost/regex/v4/states.hpp>
<boost/regex/v4/regbase.hpp>
<boost/regex/v4/iterator_traits.hpp>
<boost/regex/v4/basic_regex.hpp>
<boost/regex/v4/basic_regex_creator.hpp>
<boost/regex/v4/basic_regex_parser.hpp>
<boost/regex/v4/sub_match.hpp>
<boost/regex/v4/regex_format.hpp>
<boost/regex/v4/match_results.hpp>
<boost/regex/v4/protected_call.hpp>
<boost/regex/v4/perl_matcher.hpp>
<boost/regex/v4/instances.hpp>
<boost/regex/v4/instances.hpp>
<boost/regex/v4/instances.hpp>
<boost/regex/v4/regex_match.hpp>
<boost/regex/v4/regex_search.hpp>
<boost/regex/v4/regex_iterator.hpp>
<boost/regex/v4/regex_token_iterator.hpp>
<boost/regex/v4/regex_grep.hpp>
<boost/regex/v4/regex_replace.hpp>
<boost/regex/v4/regex_merge.hpp>
<boost/regex/v4/regex_split.hpp>

1443596553 d:\libs\include\boost-1_53\boost\regex\v4\regex_workaround.hpp

<new>
<cstring>
<cstdlib>

```

<cstddef>
<cassert>
<cstdio>
<climits>
<string>
<stdexcept>
<iterator>
<algorithm>
<iosfwd>
<vector>
<map>
<boost/limits.hpp>
<boost/assert.hpp>
<boost/cstdint.hpp>
<boost/throw_exception.hpp>
<boost/scoped_ptr.hpp>
<boost/scoped_array.hpp>
<boost/shared_ptr.hpp>
<boost/mpl/bool_fwd.hpp>
<locale>

```

1443596218 d:\libs\include\boost-1_53\boost\limits.hpp

```

<boost/config.hpp>
<boost/detail/limits.hpp>
<limits>
<climits>

```

1443596245 d:\libs\include\boost-1_53\boost\detail\limits.hpp

```

<climits>
<cfloat>
<boost/config.hpp>
<boost/detail/endian.hpp>
<wchar>

```

1443596243 d:\libs\include\boost-1_53\boost\detail\endian.hpp

```

<endian.h>

```

1443596214 d:\libs\include\boost-1_53\boost\assert.hpp

```

<boost/current_function.hpp>
<assert.h>
<boost/current_function.hpp>
<cstdlib>
<iostream>
<boost/current_function.hpp>

```

1443596216 d:\libs\include\boost-1_53\boost\stdint.hpp

```

<boost/config.hpp>

```

```

<inttypes.h>
<inttypes.h>
<stdint.h>
<inttypes.h>
<boost/limits.hpp>
<limits.h>
<limits.h>

```

```

1443596221 d:\libs\include\boost-1_53\boost\scoped_ptr.hpp
<boost/smart_ptr/scoped_ptr.hpp>

```

```

1443596343 d:\libs\include\boost-1_53\boost\smart_ptr\scoped_ptr.hpp
<boost/config.hpp>
<boost/assert.hpp>
<boost/checked_delete.hpp>
<boost/smart_ptr/detail/sp_nullptr_t.hpp>
<boost/detail/workaround.hpp>
<memory>
<boost/smart_ptr/detail/operator_bool.hpp>

```

```

1443596215 d:\libs\include\boost-1_53\boost\checked_delete.hpp

```

```

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_nullptr_t.hpp
<boost/config.hpp>
<cstddef>

```

```

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\operator_bool.hpp

```

```

1443596221 d:\libs\include\boost-1_53\boost\scoped_array.hpp
<boost/smart_ptr/scoped_array.hpp>

```

```

1443596343 d:\libs\include\boost-1_53\boost\smart_ptr\scoped_array.hpp
<boost/config.hpp>
<boost/assert.hpp>
<boost/checked_delete.hpp>
<boost/smart_ptr/detail/sp_nullptr_t.hpp>
<boost/detail/workaround.hpp>
<cstddef>
<boost/smart_ptr/detail/operator_bool.hpp>

```

```

1443596221 d:\libs\include\boost-1_53\boost\shared_ptr.hpp
<boost/smart_ptr/shared_ptr.hpp>

```

```

1443596343 d:\libs\include\boost-1_53\boost\smart_ptr\shared_ptr.hpp
<boost/config.hpp>
<boost/smart_ptr/detail/shared_ptr_nmt.hpp>
<boost/config/no_tr1/memory.hpp>

```

```

<boost/assert.hpp>
<boost/checked_delete.hpp>
<boost/throw_exception.hpp>
<boost/smart_ptr/detail/shared_count.hpp>
<boost/detail/workaround.hpp>
<boost/smart_ptr/detail/sp_convertible.hpp>
<boost/smart_ptr/detail/sp_nullptr_t.hpp>
<boost/smart_ptr/detail/spinlock_pool.hpp>
<boost/memory_order.hpp>
<algorithm>
<functional>
<typeinfo>
<cstddef>
<iosfwd>
<ostream>
<boost/smart_ptr/detail/operator_bool.hpp>

```

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\shared_ptr_nmt.hpp

```

<boost/assert.hpp>
<boost/checked_delete.hpp>
<boost/throw_exception.hpp>
<boost/smart_ptr/detail/atomic_count.hpp>
<memory>
<algorithm>
<functional>
<new>

```

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count.hpp

```

<boost/config.hpp>
<boost/smart_ptr/detail/sp_has_sync.hpp>
<boost/smart_ptr/detail/atomic_count_pthreads.hpp>
<boost/smart_ptr/detail/atomic_count_gcc_x86.hpp>
<boost/smart_ptr/detail/atomic_count_win32.hpp>
<boost/smart_ptr/detail/atomic_count_sync.hpp>
<boost/smart_ptr/detail/atomic_count_gcc.hpp>
<boost/smart_ptr/detail/atomic_count_pthreads.hpp>

```

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_has_sync.hpp

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count_pthreads.hpp

```

<pthread.h>

```

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count_gcc_x86.hpp

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count_win32.hpp

```

<boost/detail/interlocked.hpp>

```

```

1443596244 d:\libs\include\boost-1_53\boost\detail\interlocked.hpp
    <boost/config.hpp>
    <windows.h>
    <intrin.h>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count_sync.hpp
    <ia64intrin.h>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\atomic_count_gcc.hpp
    <ext/atomicity.h>
    <bits/atomicity.h>

1443596405 d:\libs\include\boost-1_53\boost\config\no_tr1\memory.hpp
    <memory>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\shared_count.hpp
    <boost/config.hpp>
    <boost/checked_delete.hpp>
    <boost/throw_exception.hpp>
    <boost/smart_ptr/bad_weak_ptr.hpp>
    <boost/smart_ptr/detail/sp_counted_base.hpp>
    <boost/smart_ptr/detail/sp_counted_impl.hpp>
    <boost/detail/workaround.hpp>
    <boost/config/no_tr1/memory.hpp>
    <functional>
    <new>
    <boost/utility/addressof.hpp>

1443596342 d:\libs\include\boost-1_53\boost\smart_ptr\bad_weak_ptr.hpp
    <exception>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base.hpp
    <boost/config.hpp>
    <boost/smart_ptr/detail/sp_has_sync.hpp>
    <boost/smart_ptr/detail/sp_counted_base_nt.hpp>
    <boost/smart_ptr/detail/sp_counted_base_spin.hpp>
    <boost/smart_ptr/detail/sp_counted_base_pt.hpp>
    <boost/smart_ptr/detail/sp_counted_base_nt.hpp>
    <boost/smart_ptr/detail/sp_counted_base_snc_ps3.hpp>
    <boost/smart_ptr/detail/sp_counted_base_gcc_x86.hpp>
    <boost/smart_ptr/detail/sp_counted_base_acc_ia64.hpp>
    <boost/smart_ptr/detail/sp_counted_base_gcc_ia64.hpp>
    <boost/smart_ptr/detail/sp_counted_base_vacpp_ppc.hpp>
    <boost/smart_ptr/detail/sp_counted_base_cw_ppc.hpp>
    <boost/smart_ptr/detail/sp_counted_base_gcc_ppc.hpp>
    <boost/smart_ptr/detail/sp_counted_base_gcc_mips.hpp>
    <boost/smart_ptr/detail/sp_counted_base_sync.hpp>

```

<boost/smart_ptr/detail/sp_counted_base_gcc_sparc.hpp>
 <boost/smart_ptr/detail/sp_counted_base_w32.hpp>
 <boost/smart_ptr/detail/sp_counted_base_aix.hpp>
 <boost/smart_ptr/detail/sp_counted_base_nt.hpp>
 <boost/smart_ptr/detail/sp_counted_base_spin.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_nt.hpp
 <boost/detail/sp_typeinfo.hpp>

1443596245 d:\libs\include\boost-1_53\boost\detail\sp_typeinfo.hpp
 <boost/config.hpp>
 <boost/current_function.hpp>
 <functional>
 <typeinfo>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_spin.hpp
 <boost/detail/sp_typeinfo.hpp>
 <boost/smart_ptr/detail/spinlock_pool.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_pool.hpp
 <boost/config.hpp>
 <boost/smart_ptr/detail/spinlock.hpp>
 <cstdint>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock.hpp
 <boost/config.hpp>
 <boost/smart_ptr/detail/sp_has_sync.hpp>
 <boost/smart_ptr/detail/spinlock_pt.hpp>
 <boost/smart_ptr/detail/spinlock_gcc_arm.hpp>
 <boost/smart_ptr/detail/spinlock_sync.hpp>
 <boost/smart_ptr/detail/spinlock_w32.hpp>
 <boost/smart_ptr/detail/spinlock_pt.hpp>
 <boost/smart_ptr/detail/spinlock_nt.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_pt.hpp
 <pthread.h>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_gcc_arm.hpp
 <boost/smart_ptr/detail/yield_k.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\yield_k.hpp
 <boost/config.hpp>
 <windows.h>
 <sched.h>
 <time.h>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_sync.hpp

<boost/smart_ptr/detail/yield_k.hpp>
<ia64intrin.h>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_w32.hpp
<boost/detail/interlocked.hpp>
<boost/smart_ptr/detail/yield_k.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\spinlock_nt.hpp
<boost/assert.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_pt.hpp
<boost/detail/sp_typeinfo.hpp>
<pthread.h>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_snc_ps3.hpp
<boost/detail/sp_typeinfo.hpp>
<inttypes.h>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_gcc_x86.hpp
<boost/detail/sp_typeinfo.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_acc_ia64.hpp
<boost/detail/sp_typeinfo.hpp>
<machine/sys/inline.h>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_gcc_ia64.hpp
<boost/detail/sp_typeinfo.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_vacpp_ppc.hpp
<boost/detail/sp_typeinfo.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_cw_ppc.hpp
<boost/detail/sp_typeinfo.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_gcc_ppc.hpp
<boost/detail/sp_typeinfo.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_gcc_mips.hpp
<boost/detail/sp_typeinfo.hpp>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_sync.hpp
<boost/detail/sp_typeinfo.hpp>
<limits.h>
<ia64intrin.h>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_gcc_sparc.hpp
<boost/detail/sp_typeinfo.hpp>

<inttypes.h>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_w32.hpp

<boost/detail/interlocked.hpp>

<boost/detail/workaround.hpp>

<boost/detail/sp_typeinfo.hpp>

1443596557 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_base_aix.hpp

<boost/detail/sp_typeinfo.hpp>

<builtins.h>

<sys/atomic_op.h>

1443596558 d:\libs\include\boost-1_53\boost\smart_ptr\detail\sp_counted_impl.hpp

<boost/config.hpp>

<boost/checked_delete.hpp>

<boost/smart_ptr/detail/sp_counted_base.hpp>

<boost/smart_ptr/detail/quick_allocator.hpp>

<memory>

<cstddef>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\quick_allocator.hpp

<boost/config.hpp>

<boost/smart_ptr/detail/lightweight_mutex.hpp>

<boost/type_traits/type_with_alignment.hpp>

<boost/type_traits/alignment_of.hpp>

<new>

<cstddef>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\lightweight_mutex.hpp

<boost/config.hpp>

<boost/smart_ptr/detail/lwm_nop.hpp>

<boost/smart_ptr/detail/lwm_threads.hpp>

<boost/smart_ptr/detail/lwm_win32_cs.hpp>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\lwm_nop.hpp

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\lwm_threads.hpp

<boost/assert.hpp>

<pthread.h>

1443596556 d:\libs\include\boost-1_53\boost\smart_ptr\detail\lwm_win32_cs.hpp

<windows.h>

1443596361 d:\libs\include\boost-1_53\boost\type_traits\type_with_alignment.hpp

<boost/mpl/if.hpp>

<boost/preprocessor/list/for_each_i.hpp>

<boost/preprocessor/tuple/to_list.hpp>

<boost/preprocessor/cat.hpp>
 <boost/preprocessor/list/transform.hpp>
 <boost/preprocessor/list/append.hpp>
 <boost/type_traits/alignment_of.hpp>
 <boost/type_traits/is_pod.hpp>
 <boost/static_assert.hpp>
 <boost/config.hpp>
 <boost/type_traits/detail/bool_trait_def.hpp>
 <cstdlibdef>
 <boost/type_traits/detail/bool_trait_undef.hpp>

1443596296 d:\libs\include\boost-1_53\boost\mpl\if.hpp

<boost/mpl/aux_/value_wknd.hpp>
 <boost/mpl/aux_/static_cast.hpp>
 <boost/mpl/aux_/na_spec.hpp>
 <boost/mpl/aux_/lambda_support.hpp>
 <boost/mpl/aux_/config/integral.hpp>
 <boost/mpl/aux_/config/ctps.hpp>
 <boost/mpl/aux_/config/workaround.hpp>

1443596489 d:\libs\include\boost-1_53\boost\mpl\aux_\value_wknd.hpp

<boost/mpl/aux_/static_cast.hpp>
 <boost/mpl/aux_/config/integral.hpp>
 <boost/mpl/aux_/config/eti.hpp>
 <boost/mpl/aux_/config/workaround.hpp>
 <boost/mpl/int.hpp>

1443596489 d:\libs\include\boost-1_53\boost\mpl\aux_\static_cast.hpp

<boost/mpl/aux_/config/workaround.hpp>

1443597540 d:\libs\include\boost-1_53\boost\mpl\aux_\config\workaround.hpp

<boost/detail/workaround.hpp>

1443597495 d:\libs\include\boost-1_53\boost\mpl\aux_\config\integral.hpp

<boost/mpl/aux_/config/msvc.hpp>
 <boost/mpl/aux_/config/workaround.hpp>

1443597539 d:\libs\include\boost-1_53\boost\mpl\aux_\config\msvc.hpp

<boost/config.hpp>

1443597495 d:\libs\include\boost-1_53\boost\mpl\aux_\config\eti.hpp

<boost/mpl/aux_/config/msvc.hpp>
 <boost/mpl/aux_/config/workaround.hpp>

1443596296 d:\libs\include\boost-1_53\boost\mpl\int.hpp

<boost/mpl/int_fwd.hpp>
 <boost/mpl/aux_/integral_wrapper.hpp>

```

1443596297 d:\libs\include\boost-1_53\boost\mpl\int_fwd.hpp
    <boost/mpl/aux_/adl_barrier.hpp>
    <boost/mpl/aux_/nttp_decl.hpp>

1443596484 d:\libs\include\boost-1_53\boost\mpl\aux_\adl_barrier.hpp
    <boost/mpl/aux_/config/adl.hpp>
    <boost/mpl/aux_/config/gcc.hpp>
    <boost/mpl/aux_/config/workaround.hpp>

1443597410 d:\libs\include\boost-1_53\boost\mpl\aux_\config\adl.hpp
    <boost/mpl/aux_/config/msvc.hpp>
    <boost/mpl/aux_/config/intel.hpp>
    <boost/mpl/aux_/config/gcc.hpp>
    <boost/mpl/aux_/config/workaround.hpp>

1443597539 d:\libs\include\boost-1_53\boost\mpl\aux_\config\intel.hpp
    <boost/config.hpp>

1443597495 d:\libs\include\boost-1_53\boost\mpl\aux_\config\gcc.hpp

1443596488 d:\libs\include\boost-1_53\boost\mpl\aux_\nttp_decl.hpp
    <boost/mpl/aux_/config/nttp.hpp>
    <boost/preprocessor/cat.hpp>

1443597540 d:\libs\include\boost-1_53\boost\mpl\aux_\config\nttp.hpp
    <boost/mpl/aux_/config/msvc.hpp>
    <boost/mpl/aux_/config/workaround.hpp>

1443596316 d:\libs\include\boost-1_53\boost\preprocessor\cat.hpp
    <boost/preprocessor/config/config.hpp>

1443596518 d:\libs\include\boost-1_53\boost\preprocessor\config\config.hpp

1443596486 d:\libs\include\boost-1_53\boost\mpl\aux_\integral_wrapper.hpp
    <boost/mpl/integral_c_tag.hpp>
    <boost/mpl/aux_/static_cast.hpp>
    <boost/mpl/aux_/nttp_decl.hpp>
    <boost/mpl/aux_/config/static_constant.hpp>
    <boost/mpl/aux_/config/workaround.hpp>
    <boost/preprocessor/cat.hpp>

```

```

}
```

ДОДАТОК В (обов'язковий)

КОПІЇ НАУКОВИХ ПУБЛІКАЦІЙ

Intelligent Information Technology for Supporting the Medical Decision-Making Considering the Legal Basis

Tetiana Hovorushchenko^a, Yelyzaveta Hnatchuk^a, Alla Herts^b and Oksana Onyshko^a

^a *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*

^b *Ivan Franko National University of Lviv, Universytetska str., 1, Lviv, 79000, Ukraine*

Abstract

Decision support systems (DSS) and information technology (IT) for medicine considering the legal basis allow free and automated verification of all essential terms of contracts and provide recommendations for the contract's conclusion or its non-conclusion. Current decisions to support medical decision-making considering the legal basis showed that none of the known decisions meet all the necessary criteria. Therefore, an actual problem for Ukraine is the development and implementation of intelligent information technology for supporting the medical decision-making considering the legal basis, which is the aim of this study. The developed intelligent information technology for supporting the medical decision-making considering the legal basis provides support for decision-making on the possibility of using reproductive technologies (possibility of surrogacy and/or in vitro fertilization), the possibility of donation and transplantation, the possibility of concluding contracts for therapeutic services and contracts for dental services, general contracts for medical services. In addition, intelligent information technology to support medical decision-making based on the legal basis automates the semantic parsing of contracts and draws conclusions about the possibility or impossibility of concluding a contract, as well as provides a request stating the reasons for impossibility to conclude a contract (for example, indicating missing essential conditions), if it was concluded that such a contract cannot be concluded.

Keywords

Decision-making, contract for the provision of medical services, conclusion about the possibility/impossibility of concluding the medical services contract, intelligent information technology for supporting the medical decision-making considering the legal basis.

1. Introduction

The implementation of the new information technologies in all human activity's areas in order to automate routine work, reduce manual labor and minimize the human factor is the strategic goal of Ukraine's information society [1]. Today, medical information technology must meet 4 requirements: efficiency, safety, novelty and economic benefits [2].

Today, decision-making processes in the field of medical services are cumbersome, complicated and non-transparent [3]. The productivity of health workers is significantly increased due to the successful implementation and use of decision support systems (DSS) and information technology (IT) [4]. Today in Ukraine there is a problem of development of medical software, IT and DSS. Successful implementation of medical software, IT and DSS provide doctors with new information on the specialty, increase the efficiency of doctors, increase the productivity of medical resources and strengthen the integration of Ukrainian medicine into the medical space of Europe [5, 6]. Intelligent or

IntellTISIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine
 EMAIL: tat_yana@ukr.net (T. Hovorushchenko); liza_veta@ukr.net (Ye. Hnatchuk); agerc@ukr.net (A. Herts); van4o@ukr.net (O. Onyshko)
 ORCID: 0000-0002-7942-1857 (T. Hovorushchenko); 0000-0003-2989-3183 (Ye. Hnatchuk); 0000-0002-3310-3159 (A. Herts); 0000-0002-5165-1810 (O. Onyshko)



intellectualized systems and technologies facilitate medical decision-making and are promising tools in the era of "evidence-based" medicine [7]. Paper [8] discusses DSS for support doctors in decision-making and for improving the health care's quality and safety. The review [9] showed that advanced clinical decision support is the prospect direction for the next 10 years. Joint decision-making involves different types of agents with the requirement that they can sufficient possibility to make the appropriate decision [10]. Intelligent IT can guide doctors in decision-making, reaching a diagnosis, and improving contract. It can reduce healthcare costs by decreasing medical errors and providing more dependable predictions [11].

The most important and complicated is the task of developing the multidisciplinary medical IT and DSS (e.g., IT and DSS for the medical law domain, because many patient health problems have a legal basis [12]), because in requirements for such IT and DSS the software development standards and standards in medicine and law need to be considered, preventing to losses of information owing to different understandings of information needs and contexts [13].

Service contracts are the most common type of contract of civil law. The contract for the medical services' provision (therapy, transplantation and donation, the reproductive technologies' use, dental services) is the most common legal basis for the medical services' provision. A contract for the medical services' provision is an agreement whereby doctor (or clinic) at the patient's request must relevantly serve the patient (restore and maintain his health) and the patient must pay a certain amount of money specified in the contract for the services provided to him. Implemented and used medical IT and DSS can substantially increase the correctness of the contract from the point of view civil and medical law [4].

The modern therapist makes decisions by integrating several medical specialties; especially this problem has been exacerbated with the introduction of insurance medicine in Ukraine, when the physician must know the standards of diagnosis and standards of treatment of therapeutic diseases. This problem is faced by therapists daily and repeatedly, especially when using innovative medical technologies [14]. Therefore, nowadays the issue of therapeutic contract's concluding needs a lot of attention. The therapeutic contract is an agreement whereby doctor at the request of the patient in accordance with health legislation provides a therapeutic service, which is creation and treatment of the etiology, pathogenesis and clinical manifestations of internal organs' diseases, their diagnostics, prophylaxis and rehabilitation, and the patient must pay a certain amount of money specified in the contract for the services provided to him, unless else provided by contract or law.

Nowadays, transplantation of human organs and (or) tissues is an effective and in many cases the only means of saving lives and restoring health. According to the Law of Ukraine "On the use of transplantation of anatomical materials to humans" [15], transplantation is a special treatment method which involves transplanting to a recipient of an organ or other anatomical material from a person or animal. Transplantation is the transfer (engraftment) of human cells, tissues or organs from the donor to the recipient in order to restore their function(s) in the body [16]. Donation is the voluntary transfer of blood and other human organs to help other people heal and recover [15]. There are 2 types of donation: posthumous (if the consent of the person given during his life or the consent of family members given after person's death is available) and lifelong if the consent of the donor is available). According to the Law of Ukraine "On the use of transplantation of anatomical materials to humans" [15], transplantation activities should be based solely on the Unified State Information System for Organ and Tissue Transplantation (USIST). Direct work with data on donors and patients in the system will be semi-automatic. According to the authors, USIST can and should be fully automated in terms of the decision-making on the possibility of donation and transplantation considering the civil law basis to minimize the influence of the human factor in making such important decisions.

Nowadays, almost 15% of Ukrainian families are infertile. The only way of overcoming infertility for many families is the reproductive technologies' using. Reproductive technologies are modern high-tech methods of infertility treatment, during of which some or all stages of conception and development of embryos carried out outside the body, in particular, fertilization of the egg outside the body, implantation of embryos and pregnancy in case of impossibility of these processes naturally [17]. The particular importance has two types of reproductive technologies - in-vitro fertilization and surrogacy. If such conditions exist, then the issue of concluding a contract for in vitro fertilization or surrogacy is very important. In vitro fertilization contract is an agreement according to which the doctor (clinic) at the patient's request must serve the patient with appropriate medical services,

applying the assisted reproductive technologies, when the eggs are fertilized with sperm outside the body, and the patient agrees to pay for the procedure, as it was agreed by the contract's parties. According to the Order of the Ministry of Health of Ukraine №787 [18], surrogacy is the artificial egg's insemination with the following placement of the embryo in another woman's uterine cavity (surrogate mother). Thus, the being of surrogacy is the transplantation of the fertilized egg into the body of a woman, which is genetically alien for future child, who bears and gives birth the child not for herself, but for a family who cannot have children for medical reasons.

Today, dental services are the popular medical service in Ukraine. Applying for dental services is second after therapeutics services [19]. Today, the issue of concluding the dental services' contract needs more attention. The dental services' contract is an agreement whereby the dentist at the patient's request, according to the health care legislation, provides a dental service, preventing the disease of the teeth, the oral cavity, gums, maxillofacial area, their diagnostics, and treatment, and the patient must pay this services if other don't specify by the dental services' contract [20].

Therefore, the conclusion of a contract for the provision of medical services, in particular, verification of the existence of all significant conditions in the contract, is becoming increasingly important - with the purpose of ensuring the safety both the doctor providing the service and the patient receiving the service, as the lack of significant conditions in the contract leads to serious problems and legal conflicts. To date, there is no single form of contract for the provision of various medical services in Ukraine. The legislative acts have no norms, which give the definition and regulate the procedure for concluding the medical services' contracts. Resulting, many medical services' contracts have aggravating conditions for the patient, offer illegal methods of dispute resolution, and have no characteristics for individualizing the medical services. These disadvantages often lead to adverse legal consequences for both the patient and the clinics, which continue to use contract models that have obvious legal limitations. Of course, every clinic cannot hire a lawyer for preparing the medical services' contracts. Multidisciplinary medical DSS and IT for the medical law domain allow free and automated verification of all essential conditions' existence in the contract and the provision of advice on the conclusion of the contract or not.

So, the development and using the intelligent information technology (set of processes, methods, and tools for accumulating, processing, and transmission of the primary information with the purpose of obtaining information product with new quality [21- 23]) for supporting the medical decision-making considering the legal basis is *an urgent problem* for Ukraine.

2. State-of-the-art

Let conduct state-of-the-art on known decisions for support of the making the medical decisions based on legal grounds. We are interested in whether the known methods and tools support decision-making on the using reproductive technologies' possibility (surrogacy and in vitro fertilization) – *criterion 1*, on the donation and transplantation possibility – *criterion 2*, on the possibility of concluding therapeutic services' contracts – *criterion 3*, dental services' contracts – *criterion 4* and general medical services' contracts – *criterion 5*. In addition, we are interested in whether known tools automate parsing the contracts and drawing conclusions about the possibility or impossibility of concluding a contract – *criterion 6*, whether these tools provide a request, what exactly is missing in the contract for its conclusion, if a conclusion about the impossibility of concluding the contract is drawing – *criterion 7*.

Results of the conducted state-of-the-art are represented in Table 1.

| | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|
| Effective method of imputation of missing data through SGTM neural-like structure [21] | no | no | no | no | no | yes | yes |
| Ontology-based approach for the information sufficiency's evaluating in the surrogate motherhood contract [35] | yes | no | no | no | no | yes | yes |
| Different models of support of the clinical ethics [36] | no | no | no | yes | yes | no | no |
| AMBOSS: Method of legal assessment to make decisions by the patient's [37] | no | no | yes | no | yes | no | no |
| Portal of Medical Data Models [38] | yes | no | yes | no | yes | yes | no |
| Information technology for filling the patients' dynamic consent [39] | no | yes | yes | yes | yes | no | no |
| Technology for legal support providing for healthcare staff [12] | no | no | no | no | no | no | yes |
| System for collection, collation, analysis, distribution and reaction for serious adverse events and reactions in unrelated HPC donors [40] | no | yes | no | no | no | no | no |
| Global Database on Donation and Transplantation [41] | no | yes | no | no | no | yes | no |
| Kidney Allocation System [42] | no | yes | no | no | no | yes | no |
| Korean Organ Transplantation Registry (KOTRY) [43] | no | yes | no | no | no | yes | no |
| Donor Advocacy Team (DAT) program [44] | no | yes | no | no | no | yes | no |
| National Subsystem of donation and transplantation in Mexico [45] | no | yes | no | no | no | yes | no |
| Polish organ transplantation management system [46] | no | yes | no | no | no | yes | no |
| French national information system for coordination and administration of organ transplant [47] | no | yes | no | no | no | yes | no |
| Intelligent agent for support of decision making on civil law regulation of contract for the provision of in vitro fertilization [48] | yes | no | no | no | no | yes | yes |
| Decision-making about conclusion of medical services' contractual obligations [49] | no | no | no | no | yes | yes | yes |
| Information technology for the dental services contract's legal | no | no | no | yes | no | yes | yes |

| | | | | | | | | |
|--|----|-----|-----|----|----|-----|-----|--|
| regulation [50] | | | | | | | | |
| Supporting the decision-making about the donation's and transplantation's possibility considering the civil law grounds [51] | no | yes | no | no | no | yes | yes | |
| Decision making process on civil law regulation of contracts for the provision of therapeutic services [52] | no | no | yes | no | no | yes | yes | |

The conducted state-of-the-art on known decisions for support of the making the medical decisions based on legal grounds showed, none of the known solutions satisfies all 7 criteria in the complex. Therefore, developing and implementing intelligent information technology for supporting the medical decision-making considering the legal basis, which would meet all the above 7 criteria simultaneously is *the purpose of this study*.

3. Intelligent information technology for supporting the medical decision-making considering the legal basis

On the basis of the modeling the information flows in the medical decision-making process considering the legal basis and the investigation of information flows in the process of evaluating information, which is available for medical decision-making considering the legal basis, conducted by the authors in [21, 48-52], and theoretical principles of information technology in the form of the methods for decision-making support, which were presented in [21, 48-52], the block diagram of intelligent information technology for supporting the medical decision-making considering the legal basis (set of processes, methods, and tools for accumulating, processing, and transmission of the primary information with the purpose of obtaining information product with new quality [21, 22]) is developed and presented in Figure 1.

Intelligent information technology for supporting the medical decision-making considering the legal basis gives the support for decision-making on the possibility of using reproductive technologies (the possibility of surrogacy and/or in vitro fertilization), the possibility of donation and transplantation, the possibility of concluding therapeutic services' contracts, dental services' contracts and general medical services' contracts. In addition, intelligent information technology for supporting the medical decision-making considering the legal basis automates the semantic parsing of contracts and drawing conclusions about the possibility or impossibility of concluding a contract, as well as provides a request indicating the reasons for impossibility of concluding a contract (for example, indicating the missing significant conditions in the contract), if it was concluded that it is impossible to conclude such a contract.

Figure 1 showed that the proposed intelligent information technology for supporting the medical decision-making considering the legal basis consists of:

1. Parsing the appropriate medical services' contract (for example, using an intelligent agent for parsing natural language documents, which was developed in [53])
2. Method of decision-making in the field of legal regulation of surrogacy [21]
3. Method of decision making on civil law regulation of contract for the provision of in-vitro fertilization [48]
4. Method for supporting the decision on the possibility of donation and transplantation on the basis of civil law [51]
5. Method of decision making process on civil law regulation of contracts for the provision of therapeutic services [52]
6. Method for legal regulation of the dental services contract [50]
7. Method of decision-making about conclusion of contractual obligations in the field of medical services [49]

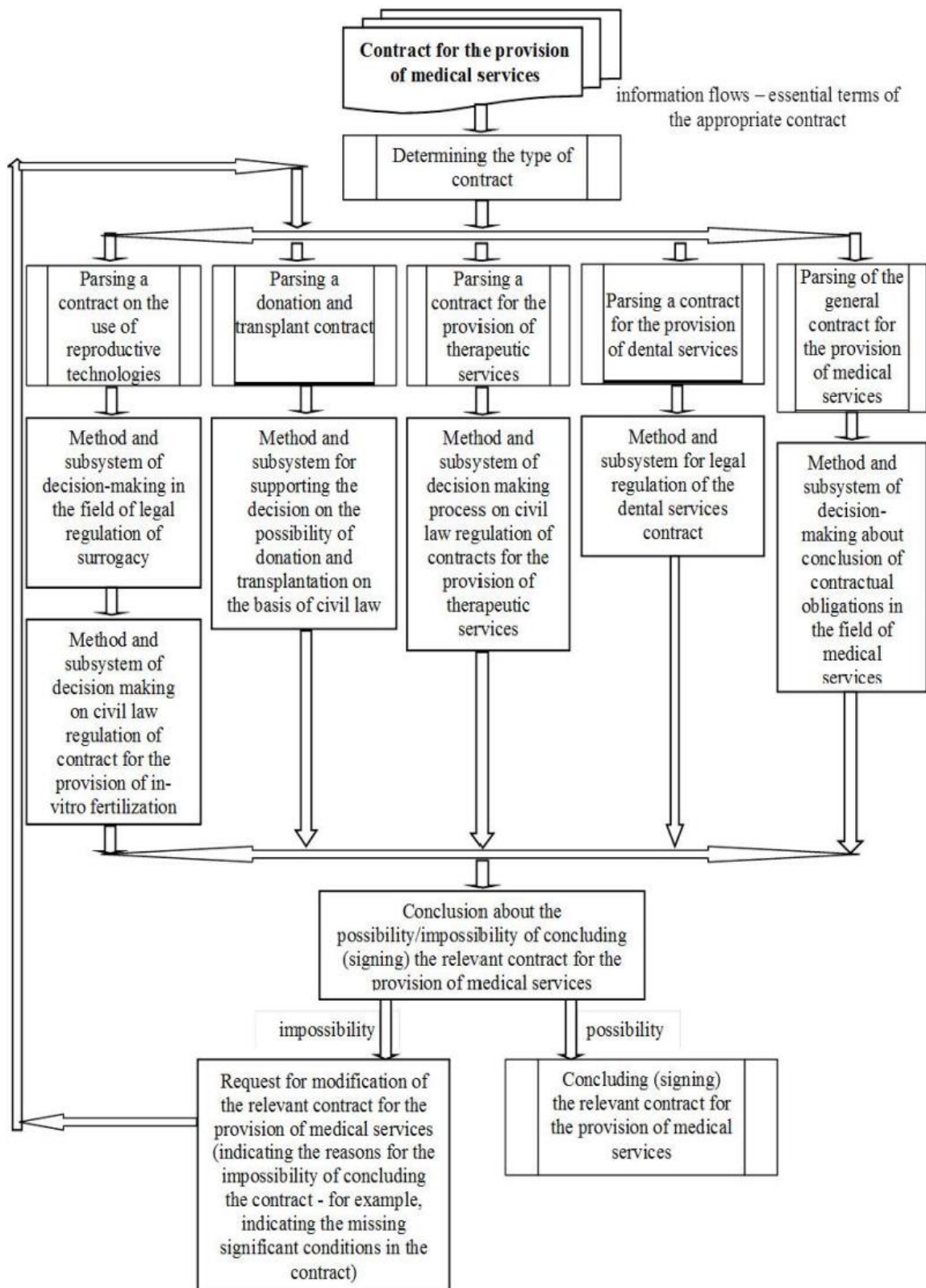


Figure 1: Intelligent information technology for supporting the medical decision-making considering the legal basis

Obviously, the main purpose of the developed intelligent information technology for supporting the medical decision-making considering the legal basis is the automation (minimization of the human factor impact and simplification of the implementation) of the contracts' parsing and drawing the conclusions about the possibility or impossibility of concluding the contracts of reproductive technologies (surrogacy and/or in vitro fertilization), of donation and transplantation, therapeutic services' contract, dental services' contract, and general medical services' contract.

4. Results & discussion

Let consider the functioning of the developed intelligent information technology for supporting the medical decision-making considering the legal basis.

For example, the input of the developed information technology is the medical services' contract. It was determined that this is a contract for the provision of in-vitro fertilization. Developed intelligent information technology for supporting the medical decision-making considering the legal basis conducted a semantic analysis (parsing) of the provided contract for the provision of in-vitro fertilization, as well as processing the contract using the method of decision making on civil law regulation of the contract for the provision of in-vitro fertilization, which was developed by the authors in [48]. Resulting in the analysis of the contract, a set of significant conditions, which are absent in the analyzed contract, was formed. The set of missing significant conditions has the form: $MC = \{ \text{"indications for in-vitro fertilization"}, \text{"woman's age"}, \text{"woman's legal capacity"} \}$. It is not empty, so intelligent information technology for supporting the medical decision-making considering the legal basis forms conclusion on the impossibility of concluding the appropriate medical services' contract, and a request for addition of the in-vitro fertilization contract (indicating the absence of significant conditions in the contract). Thus, the developed information technology helped to avoid concluding the incorrectly executed contract, which have no significant conditions, and could lead to negative effects for one or both parties.

The developed intelligent information technology for supporting medical decision-making considering the legal basis gives the opportunity to verify the correctness of the structure and content of the appropriate medical services' contract. Such verification allows the unmistakable (from the point of view of legal basis) decisions on the possibility/impossibility of concluding the contract, minimizing the subjectivism influence and the human factor, and eliminating the need to pay for legal services by the clinics.

5. Conclusions

The conclusion of the medical services' contract, in particular, verification of the availability of all significant conditions in the contract, is becoming increasingly important, with the purpose of ensuring the safety of both the doctor providing the service and the patient receiving the service, as the lack of significant conditions in the contract leads to serious problems and legal conflicts. Many contracts for the provision of medical services of various kinds contain aggravating conditions for the patient, offer non-legal method of dispute resolution, and do not contain characteristics that individualize medical services. These shortcomings can lead to unfavorable legal consequences for both the patient and the medical organizations, which continue to use such models of contracts, which have obvious restrictions on the legal side. Of course, not every clinic can afford to hire a lawyer to prepare contracts for the provision of medical services. Multidisciplinary medical DSS and IT for the field of medical law are designed to provide the possibility of free and automated verification of the existence of all significant conditions in the contract and provide advice on whether or not to conclude the contract. The conducted state-of-the-art on known decisions for support of the making the medical decisions based on legal grounds showed, that none of the known solutions meets all the necessary criteria in the complex. Therefore, the current problem for Ukraine is the design and implementation of intelligent information technology for supporting the medical decision-making considering the legal basis, which is why this study is devoted.

The proposed intelligent information technology for supporting the medical decision-making considering the legal grounds provides support for decision-making on the possibility of using

reproductive technologies (on the possibility of surrogacy and/or in-vitro fertilization), on the possibility of donation and transplantation, on the possibility of concluding contracts for therapeutic services, contracts for dental services and general contracts for medical services. In addition, intelligent information technology for supporting the medical decision-making considering the legal basis automates the semantic parsing of contracts and drawing conclusions about the possibility or impossibility of concluding a contract, as well as provides a request indicating the reasons for impossibility of concluding a contract (for example, indicating the missing significant conditions in the contract), if it was concluded that it is impossible to conclude such a contract.

6. References

- [1] T. Hovorushchenko, O. Pavlova, M. Bodnar, Development of an intelligent agent for analysis of nonfunctional characteristics in specifications of software requirements. *Eastern-European Journal of Enterprise Technologies* 1 2 (2019) 6-17. doi: 10.15587/1729-4061.2019.154074.
- [2] V. Denysyuk, O. Denysyuk, Evidence-Based Internal Medicine, SE "State Cartographic Factory", Vinnytsya, 2011.
- [3] C. Adams, J. Allen, F. Flack, Data custodians and the decision-making process: releasing data for research. *Journal of Law and Medicine* 26 2 (2018) 433-453.
- [4] Y. Syerov, N. Shakhovska, S. Fedushko, Method of the data adequacy determination of personal medical profiles. *Advances in Intelligent Systems and Computing* 902 (2020) 333-343. doi: 10.1007/978-3-030-12082-5_31.
- [5] N. Melnykova, N. Shakhovska, T. Sviridova, The personalized approach in a medical decentralized diagnostic and treatment, in: *Proceedings of the 14-th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics CADSM-2017*, Lviv, 2017, pp. 295-297. doi: 10.1109/CADSM.2017.7916139.
- [6] O. Berezhsky, M. Zarichnyi, O. Pitsun, Development of a metric and the methods for quantitative estimation of the segmentation of biomedical images. *Eastern-European Journal of Enterprise Technologies* 6 4 (2017) 4-11. doi: 10.15587/1729-4061.2017.119493.
- [7] C. Mitchell, C. Ploem, Legal challenges for the implementation of advanced clinical digital decision support systems in Europe. *Journal of Clinical and Translational Research* 3 (2018) 424-430. doi: 10.18053/jctres.03.2017S3.005.
- [8] K. Cresswell, A. Majeed, D. Bates, A. Sheikh, Computerised decision support systems for healthcare professionals: An interpretative review. *The Journal of Innovation in Health Informatics* 20 2 (2012) 115-128. doi: 10.14236/jhi.v20i2.32.
- [9] M. Collen, M. Ball, Medical informatics: past and future. *History of medical informatics in the United States* (2015) 725-748. doi: 10.1007/978-1-4471-6732-7_18.
- [10] K. Knight, Who is the patient? Tensions between advance care planning and shared decision-making. *Journal of Evaluation in Clinical Practice* 25 6 (2019) 1217-1225. doi: 10.1111/jep.13149.
- [11] P. Iftikhar, M. Kuijpers, A. Khayyatt, A. Iftikhar, M. De Sa, Artificial intelligence: A new paradigm in obstetrics and gynecology research and clinical practice. *Cureus* 12 2 (2020), e7124. doi: 10.7759/cureus.7124.
- [12] S. Agate, M. Curran, Opportunity for legal innovation in healthcare technology, 2018. URL: <https://www.lawpracticetoday.org/article/legal-innovation-healthcare-technology>.
- [13] T. Hovorushchenko, O. Pomorova, Evaluation of Mutual Influences of Software Quality Characteristics Based ISO 25010:2011, in: *Proceedings of 2016 International Scientific and Technical Conference "Computer Science and Information Technologies", CSIT'2016*, Lviv, 2016, pp. 80-83. doi: 10.1109/STC-CSIT.2016.7589874.
- [14] Y. Khrustalev, Therapy is the basis for strengthening the health of the nation. *Therapeutic archive*. 8 (2018) 78-80.
- [15] Law of Ukraine "On the use of transplantation of anatomical materials to humans". Information of the Verkhovna Rada of Ukraine (2018).
- [16] World Health Organization: Transplantation, 2020. URL: <https://www.who.int/topics/transplantation/en>.

- [17] R. Maydanyk, *Reproductive rights. Surrogate motherhood*, Alerta, Kyiv, 2013.
- [18] Order of the Ministry of Health "On Approval of the Procedure for the Use of Assisted Reproductive Technologies in Ukraine No. 787 of 09.09.2013". *Official Journal of Ukraine*. 82 (2013) 446.
- [19] V. Bardov, *The Basics of Dental Activity (Organizational-Legal, Hygienic, Deontological)*. Nova Knyga, Vinnytsya, 2011.
- [20] Civil Code of Ukraine. Information of the Verkhovna Rada of Ukraine 40-44 (2003).
- [21] T. Hovorushchenko, A. Herts, Ye. Hnatchuk, Concept of intelligent decision support system in the legal regulation of the surrogate motherhood. *CEUR-WS 2488* (2019) 57-68.
- [22] O. Pomorova, T. Hovorushchenko, Artificial neural network for software quality evaluation based on the metric analysis, in: *Proceedings of IEEE East-West Design & Test Symposium, EWDTs'2013, Kharkiv, 2013*, pp.200-203. doi: 10.1109/EWDTs.2013.6673193.
- [23] T. Korkishko, A. Melnyk, Cryptographic processor architectures for DES algorithm, in: *Proceedings of IEEE Africon. 5th Africon Conference in Africa, Cape Town, 1999*, vol. 1, pp. 175-180. doi: 10.1109/AFRCON.1999.820788.
- [24] S. Khodambashi, J. Gulla, P. Abrahamsson, F. Moser, Design and development of a mobile decision support system: guiding clinicians regarding law in the practice of psychiatry in emergency department, in: *Proceedings of IEEE 30-th International Symposium on Computer-Based Medical Systems CBMS-2017, Thessaloniki, 2017*, pp. 67-72. doi: 10.1109/CBMS.2017.77.
- [25] I. Khan, M. Sher, J. Khan, S. Saqlain, A. Ghani, H. Naqvi, M. Ashraf, Conversion of legal text to a logical rules set from medical law using the medical relational model and the world rule model for a medical decision support system. *Informatics-Basel* 3 1 (2016) 2. doi: 10.3390/informatics3010002.
- [26] J. Varghese, S. Schulze Suenninghausen, M. Dugas, Standardized quality assurance forms for organ transplantations with multilingual support, open access and UMLS coding. *Studies in Health Technology and Informatics* 212 (2015) 15-22.
- [27] E. Rakus-Andersson, L. Jain, Computational intelligence in medical decisions making. *Recent Advances in Decision Making* 222 (2009) 145-159. doi: 10.1007/978-3-642-02187-9_9.
- [28] A. Nisal, U. Diwekar, V. Bhalerao, Personalized medicine for in vitro fertilization procedure using modeling and optimal control. *Journal of Theoretical Biology* 487 (2020) 110105. doi: 10.1016/j.jtbi.2019.110105.
- [29] C. Chan, B. Lau, M. Tam, E. Ng, Preferred problem solving and decision-making role in fertility treatment among women following an unsuccessful in vitro fertilization cycle. *BMC Women Health* 19 1 (2019) 153. doi: 10.1186/s12905-019-0856-5.
- [30] B. Lunenfeld, W. Bilger, S. Longobardi, J. Kirsten, T. D'Hooghe, S. Sunkara, Decision points for individualized hormonal stimulation with recombinant gonadotropins for treatment of women with infertility. *Gynecological Endocrinology* 35 12 (2019) 1027-1036. doi: 10.1080/09513590.2019.1650345.
- [31] P. Delbon, The protection of health in the care and trust relationship between doctor and patient: Competence, professional autonomy and responsibility of the doctor and decision-making autonomy of the patient. *Journal of Public Health Research* 7 3 (2018) 97-100. doi: 10.4081/jphr.2018.1423.
- [32] R. Bouvet, The primacy of the patient's wishes in the medical decision-making procedure established by French law. *European Journal of Health Law* 25 4 (2018) 426-440. doi: 10.1163/15718093-12540384.
- [33] P. Stankova, J. Horkelova, J. Luczewska, J. Ticha, S. Zimcikova, J. Cernobila, The key factors influencing clients' decision-making in the market of selected planned healthcare in the Czech Republic. *Journal of Competitiveness* 9 4 (2017) 94-113. doi: 10.7441/joc.2017.04.07.
- [34] N. Davoody, S. Koch, I. Krakau, M. Hagglund, Accessing and sharing health information for post-discharge stroke care through a national health information exchange platform - a case study. *BMC Medical Informatics and Decision Making* 19 (2019) 95. doi: 10.1186/s12911-019-0816-x.
- [35] O. Mishchuk, R. Tkachenko, I. Izonin, Missing data imputation through SGTm neural-like structure for environmental monitoring tasks. *Advances in Intelligent Systems and Computing* 938 (2019) 142-151. doi: 10.1007/978-3-030-16621-2_13.

- [36] A. Slowther, C. Johnston, J. Goodall, T. Hope, *A practical guide for clinical ethics support*, Ethox Centre, Oxford, 2004.
- [37] AMBOSS: Principles of medical law and ethics, 2020. URL: https://www.amboss.com/us/knowledge/Principles_of_medical_law_and_ethics.
- [38] M. Dugas, P. Neuhaus, A. Meidt, J. Doods, M. Storck, P. Bruland, J. Varghese, Portal of medical data models: information infrastructure for medical research and healthcare. Database: The Journal of Biological Databases and Curation (2016) bav121. doi: 10.1093/database/bav121.
- [39] Y. Joly, B. Knoppers, *Routledge Handbook of Medical Law and Ethics*, Routledge, New York, 2015.
- [40] B. Shaw, J. Chapman, M. Fechter, L. Foeken, H. Greinix, W. Hwan, L. Philips-Johnson, M. Korhonen, B. Lindberg, W. Navarro, Towards a global system of vigilance and surveillance in unrelated donors of haematopoietic progenitor cells for transplantation. *Bone Marrow Transplantation* 48 12 (2013) 1506-1509. doi: 10.1038/bmt.2013.104.
- [41] R. Matesanz, B. Mahillo, M. Alvarez, M. Carmona, Global observatory and database on donation and transplantation: World overview on transplantation activities. *Transplantation Proceedings* 41 6 (2009) 2297-2301. doi: 10.1016/j.transproceed.2009.05.004.
- [42] C. King, J. Perkins, C. Johnson, C. Blosser, N. Leca, L. Sibulesky, Utilization of standard criteria donor and expanded criteria donor kidneys after kidney allocation system implementation. *Annals of Transplantation* 23 (2018) 691-703. doi: 10.12659/AOT.910504.
- [43] J. Yang, J. Jeong, J. Lee, Y. Kim, H. Paik, J. Kim, H. Park, M. Kim, Design and methods of the Korean organ transplantation registry. *Transplantation Direct* 3 8 (2017) e191. doi: 10.1097/TXD.0000000000000678.
- [44] S. Eguchi, A. Soyama, K. Nagai, Y. Miyazaki, S. Kurihara, M. Hidaka, S. Ono, T. Adachi, K. Natsuda, T. Hara, The donor advocacy team: a risk management program for living organ, tissue, and cell transplant donors. *Surgery Today* 47 8 (2017) 980-985. Doi: 10.1007/s00595-017-1468-z.
- [45] S. Tenorio, L. Daza, S. Estrada, Electronic tool for distribution and allocation of heart on donation and transplantation in Mexico. *Transplantation Proceedings* 48 2 (2016) 564-567. doi: 10.1016/j.transproceed.2016.03.018.
- [46] J. Czerwinski, P. Kalicinski, R. Danielewicz, Serious adverse events and reactions in organ donation and transplantation: A web-net tool-based nationwide system for reporting and monitoring. *Annals of Transplantation* 20 (2015) 243-248. doi: 10.12659/AOT.893009.
- [47] W. Strang, P. Tuppin, A. Atinault, C. Jacquelinet, The French Organ Transplant Data System. *Connecting Medical Informatics and Bio-Informatics* 116 (2005) 77-82.
- [48] T. Hovorushchenko, A. Herts, Ye. Hnatchuk, Intelligent agent for support of decision making on civil law regulation of contract for the provision of in vitro fertilization, in: *Proceedings of IEEE International Scientific and Technical Conference "Computer Science and Information Technologies"*, CSIT-2020, Lviv-Zbarazh, 2020, vol. 1, pp. 312-315. doi: 10.1109/CSIT49958.2020.9322023.
- [49] T. Hovorushchenko, Ye. Hnatchuk, A. Herts, Decision-making about conclusion of contractual obligations in the field of medical services. *CEUR-WS* 2753 (2020) 142-148.
- [50] T. Hovorushchenko, A. Herts, Ye. Hnatchuk, Information technology for legal regulation of the dental services contract. *CEUR-WS* 2623 (2020) 14-24.
- [51] T. Hovorushchenko, A. Herts, Ye. Hnatchuk, O. Sachenko, Supporting the decision-making about the possibility of donation and transplantation based on civil law grounds. *Advances in Intelligent Systems and Computing* 1246 (2021) 357-376. doi: 10.1007/978-3-030-54215-3_23.
- [52] T. Hovorushchenko, A. Herts, Ye. Hnatchuk, Modeling the decision making process on civil law regulation of contracts for the provision of therapeutic services. *CEUR-WS* 2711 (2020) 333-342.
- [53] O. Pavlova, T. Hovorushchenko, A. Boyarchuk. Method of activity of intelligent agent for semantic analysis of software requirements, in: *Proceedings of IEEE 10-th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS-2019, Mez, 2019, vol.2, pp. 902-906. doi: 10.1109/IDAACS.2019.8924292.

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XIII Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2021»

15-16 жовтня 2021

Хмельницький 2021

| | |
|---|-----|
| Левчик Т. С., Собко О. В., Житкевич В. В., Міхалевський В. Ц. Метод автоматизованого діагностування хвороб рослинних культур..... | 359 |
| Манзюк Е. А., Скрипник Т. К. Система цільової кластеризації на послідових даних..... | 364 |
| Матвійчук І. І., Багрій Р. О., Скрипник Т. К. Моделювання web-орієнтованих систем | 367 |
| Мельник В. С., Міхалевський В. Ц., Скрипник Т. К. Інформаційна система для комплексної обробки деревини..... | 372 |
| Огнєвий О. В., Медведчук В. Ю., Медведчук Н. К. Основні принципи організації і особливості відеоконференцзв'язку | 375 |
| Онишко О. Г. Метод та програмні засоби препроцесінгу вхідного текстового контенту | 379 |
| Radiuk P. M. A mental model approach for making decisions in it project management | 381 |
| Пасічник О. А. Програмна система методу вимірювання лінійних переміщень за аналізом зображень..... | 385 |
| Павловський В. І., Савосько О. М. Виявлення шкідливого трафіку за використанням глибинного навчання..... | 390 |
| Пасічник О. А., Ющенко В. Б., Скрипник Т. К. Інформаційні технології як засіб автоматизації та оптимізації маркетингових кампаній в соціальних мережах..... | 395 |
| Петровський С. С. Метод зваженої оцінки успішності навчання у школі..... | 398 |
| Рожков Д. В., Петровський С. С., Скрипник Т. К. Інформаційна система організації обігу нормативних документів | 401 |
| Скрипник Т. К., Манзюк Е. А. Метод машинного навчання для визначення якості перекладу текстової інформації..... | 404 |
| Ющенко В. Б., Скрипник Т. К., Пасічник О. А. Інформаційні технології у соц-медіа: PR, реклама, лідогенерація | 406 |
| Яковчук М. В., Міхалевський В. Ц., Скрипник Т. К. Система прийняття рішень у виробничих процесах сільськогосподарського підприємства. | 408 |
| Яшина О. М., Мартинюк О. Р. Система управління якістю у розробці програмних продуктів | 410 |

УДК 004.4

Онишко О. Г.

*Хмельницький національний університет***МЕТОД ТА ПРОГРАМНІ ЗАСОБИ ПРЕПРОЦЕСИНГУ ВХІДНОГО
ТЕКСТОВОГО КОНТЕНТУ**

Проведено аналіз та рекомендується до розробки метод та програмні засоби препроцесингу вхідного текстового коду. Для успішного завдання ідентифікації текстів використовується комп'ютерний лінгвістичний аналіз текстів. Проаналізовано методи синтаксичного та семантичного аналізу та розроблено новий підхід у вигляді методу семантико-синтаксичного аналізу

Успішне розв'язання задач ідентифікації, прогнозування та діагностики неможливе без попередньої формалізації задачі та підготовки даних тобто їх препроцесингу. Тому використовуються різні види аналізу природної мови. Однією із перспективних галузей яка розглядає ці задачі є комп'ютерний лінгвістичний аналіз.

Комп'ютерний лінгвістичний аналіз текстів природної мови є галуззю штучного інтелекту, що швидко розвивається. Одним з ключових завдань обчислювального лінгвістичного аналізу є побудова такого структурованого представлення тексту, до якого можуть бути застосовані методи і алгоритми вирішення прикладних задач. Багато дослідників відзначають, що якісне вирішення таких задач, як пошук на основі питань, вилучення інформації і знань з текстів, автоматичне абстрагування, неможливе без проведення синтаксичного і семантичного аналізу.

Існує значна кількість різновидів методів як синтаксичного, так і семантичного аналізу, які засновані на різних моделях синтаксичної структури речення і різним розумінням семантики.

Метою роботи є підвищення якості автоматичного аналізу текстів на природній мові на основі інтеграції методів синтаксичного та семантичного аналізу

Дерева залежності моделюють синтаксичну структуру речень як ієрархію слів, з'єднаних дугами, що позначають синтаксичне підпорядкування між головним і залежним словами. Підпорядкування обумовлено набором загальних принципів, які зводяться до того, що залежне слово в реченні є уточнююче, необов'язкове, менш важливе для передачі значення твердження, ніж головне.

Семантика моделі, заснована на рольовій структурі речення, дозволяє абстрагуватися від синтаксичних дерев речень і порівнювати одні і ті ж семантичні

структури з різними граматичними конструкціями. Завдання визначення рольових структур тверджень включає пошук предикатних слів, які описують певну ситуацію в реченні, пошук їх семантичних аргументів - синтаксичних конструкцій, які виражають у реченні учасників ситуації, а також визначення значень аргументів, тобто присвоєння їм семантичних ролей, які грають учасники в ситуації.

Проблеми синтаксичного і семантичного аналізу, як правило, вирішуються окремо: спочатку будується синтаксичну структура речення, на основі якої потім будується семантична структура. Для цього існує ряд методів, заснованих як на правилах, так і на машинному навчанні. Хоча сучасні методи дозволяють домогтися досить якісного вирішення цих проблем, залишається значний простір для вдосконалення цих методів. Аналіз помилок синтаксичних і семантичних аналізаторів показує, що, з одного боку, для побудови правильної синтаксичної структури речення необхідно знання його семантики, а з іншого боку, помилки в дереві синтаксису негативно впливають на якість семантичного аналізу. Виходячи з цього, було проаналізовано, що поєднання синтаксичним і семантичними типами аналізу може поліпшити їх якість. Такий комбінований підхід до вирішення проблеми глибокого лінгвістичного аналізу буде називатися семантично-синтаксичним аналізом.

ISSN 2307-5732

DOI 10.31891/2307-5732

НАУКОВИЙ ЖУРНАЛ

6.2020

ВІСНИК

**Хмельницького
національного
університету**

Технічні науки

Technical sciences

SCIENTIFIC JOURNAL

HERALD OF KHMELNYTSKYI NATIONAL UNIVERSITY

2020, Issue 6, Volume 291

Хмельницький

**ВІСНИК
ХМЕЛЬНИЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
серія: Технічні науки**

Затверджений як фахове видання категорії «Б»,
РІШЕННЯ АТЕСТАЦІЙНОЇ КОЛЕГІЇ № 1643 ВІД 28.12.2019 та №409 від 17.03.2020

Засновано в липні 1997 р.

Виходить 6 разів на рік

Хмельницький, 2020, № 6(291)

**Засновник і видавець: Хмельницький національний університет
(до 2005 р. – Технологічний університет Поділля, м. Хмельницький)**

Включено до науково-метричних баз:

| | |
|--------------------------------------|---|
| Google Scholar | http://scholar.google.com.ua/citations?hl=uk&user=aLUP9OYAAAAJ |
| Index Copernicus | http://jml2012.indexcopernicus.com/passport.php?id=4538&id_lang=3 |
| Polish Scholarly Bibliography | https://pbn.nauka.gov.pl/journals/46221 |

| | |
|--------------------------------------|---|
| Головний редактор | Скиба М. Є. , д.т.н., професор, заслужений працівник народної освіти України, член-кореспондент Національної академії педагогічних наук України, ректор Хмельницького національного університету |
| Заступник головного редактора | Синюк О. М. , д.т.н., професор кафедри машин і апаратів, електромеханічних та енергетичних систем Хмельницького національного університету |
| Відповідальний секретар | Горященко С. Л. , к.т.н., доцент кафедри машин і апаратів, електромеханічних та енергетичних систем Хмельницького національного університету |

Ч л е н и р е д к о л е г і ї

Технічні науки

Березненко С.М., д.т.н., Бойко Ю.М., д.т.н., Говорущенко Т.О., д.т.н., Гордєєв А.І., д.т.н., Грабко В.В., д.т.н., Диха О.В., д.т.н., Защепкіна Н.М., д.т.н., Захаркевич О.В., д.т.н., Злотенко Б.М., д.т.н., Зубков А.М., д.т.н., Каплун П.В., д.т.н., Карташов В.М., д.т.н., Кичак В.М., д.т.н., Мазур М.П., д.т.н., Мандзюк І.А., д.т.н., Мартинюк В.В., д.т.н., Мельничук П.П., д.т.н., Місяць В.П., д.т.н., Мясіщев О.А., д.т.н., Нелін Є.А., д.т.н., Павлов С.В., д.т.н., Параска О.А., к.т.н., Рогатинський Р.М., д.т.н., Горошко А.В., д.т.н., Сарібекова Д.Г., д.т.н., Семенко А.І., д.т.н., Славінська А.Л., д.т.н., Харжевський В.О., д.т.н., Шинкарук О.М., д.т.н., Шклярський В.І., д.т.н., Щербань Ю.Ю., д.т.н., Ясній П.В., д.т.н., професор, Бубулєс Альгімантас, доктор наук (Литва), Елсаєд Ахмед Ельнашар, доктор наук (Єгипет), Кальчинські Томаш, доктор наук (Польща), Коробко Євгенія Вікторівна, д.т.н. (Білорусія), Лунтовський Андрій Олегович, д.т.н. (Німеччина), Матушевський Мацей, доктор наук (Польща), Мушлевський Лукаш, доктор наук (Польща), Мушял Януш, доктор наук (Польща), Натріашвілі Тамаз Мамієвич, д.т.н., (Грузія), Попов Валентин, доктор природничих наук (Німеччина)

| | |
|---------------------------|-------------------------|
| <i>Технічний редактор</i> | Горященко К. Л., к.т.н. |
| <i>Редактор-коректор</i> | Броженко В. О. |

**Рекомендовано до друку рішенням вченої ради Хмельницького національного університету,
протокол № 6 від 30.12.2020 р.**

Адреса редакції: редакція журналу "Вісник Хмельницького національного університету"
Хмельницький національний університет
вул. Інститутська, 11, м. Хмельницький, Україна, 29016

| | | | |
|----------------|--|-------------|---|
| ☎ | (038-2) 67-51-08 | web: | http://journals.khnu.km.ua/vestnik |
| e-mail: | visnyk.khnu@khmnu.edu.ua visnyk.khnu@gmail.com | | http://lib.khnu.km.ua/visnyk_tup.htm |

Зареєстровано Міністерством України у справах преси та інформації.
Свідоцтво про державну реєстрацію друкованого засобу масової інформації
Серія КВ № 9722 від 29 березня 2005 року

© Хмельницький національний університет, 2020
© Редакція журналу "Вісник Хмельницького національного університету", 2020

ЗМІСТ

**КОМП'ЮТЕРНІ НАУКИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІ,
СИСТЕМНИЙ АНАЛІЗ ТА КІБЕРБЕЗПЕКА**

| | |
|---|----|
| О.С. АНДРОЩУК, О.В. НАГРЕБЕЦЬКИЙ, В.С. ОРЛЕНКО, В.М. ЧЕШУН, А.І. КАТАЄВА БАЗОВІ ОПЕРАЦІЇ АЛГОРИТМУ ФОРМУВАННЯ ШИФРІВ ЗСУВУ ІЗ ЗАСТОСУВАННЯМ ЕНТРОПІЙНОГО КОДУВАННЯ ХАФФМЕНА | 7 |
| Б. В. ІВАНЩЕВ, А. В. КАПЛУНОВ, В.В. РУСІНОВ, А. М. ВОЛОКИТА МЕТОД ПРИСКОРЕННЯ ВІДНОВЛЕННЯ КОЕФІЦІЄНТУ РЕПЛІКАЦІЇ В РОЗПОДІЛЕНОМУ СХОВИЩІ ДАНИХ НА ОСНОВІ МОНІТОРИНГУ РІВНЯ ДОВІРИ ДО ВУЗЛА | 13 |
| А.С. КАШТАЛЬЯН, О.С. САВЕНКО КЛАСТЕРИЗАЦІЯ ЗЛОВМИСНИКІВ У КОМП'ЮТЕРНИХ МЕРЕЖАХ ЗА ЇХ ПОВЕДІНКОВИМИ ХАРАКТЕРИСТИКАМИ | 22 |
| П.М. ШПАТАР, О.В. ГРЕСЬ, В.В. КАЧУР, А.Я. ТОМУЛЕЦЬ ДЕТЕКТУВАННЯ ПООДИНОКИХ ФОТОНІВ В КВАНТОВИХ КРИПТОГРАФІЧНИХ СИСТЕМАХ ... | 28 |
| В.М. ДЖУЛІЙ, І.В. МУЛЯР, В.С. ОРЛЕНКО, В.Ю. ТІТОВА, В.А. АНІКІН СИМЕТРИЧНА КРИПТОСИСТЕМА З НЕЛІНІЙНИМ ШТФРУВАННЯМ ТА МОЖЛИВІСТЮ КОНТРОЛЮ ШИФРОТЕКСТУ З МЕТОЮ МАСКУВАННЯ | 33 |
| О.М. ШИНКАРУК, О.М. ЯШИНА, О.Г. ОНИШКО УПРАВЛІННЯ ЯКІСТЮ ПРОГРАМНИХ ВЕБ-СИСТЕМ ЗАСОБАМИ РОЗРОБКИ | 39 |
| Ю.І. ТИХОНОВ МІЖНАРОДНІ ПОКАЗНИКИ РІВНЯ ОСВІТИ | 45 |

ЕКОЛОГІЯ

| | |
|---|----|
| Є.Г. ДУДНИК АНАЛІЗ ВПЛИВУ ТЕХНОГЕННИХ ФАКТОРІВ НА ДЕНДРОФОРУ М. ВІННИЦІ | 51 |
|---|----|

МАШИНОБУДУВАННЯ, МЕХАНІКА ТА МАТЕРІАЛОЗНАВСТВО

| | |
|--|----|
| М. М. КОСПОК, А. М. КОСПОК, В. С. КРАВЧУК ПРОЕКТУВАННЯ ОБОРОТНОГО ПЕРЕТВОРЮВАЧА НАПРЯМКУ РУХУ | 56 |
| М.П. МАЗУР РОЗРАХУНОК ТЕМПЕРАТУРНОГО ПОЛЯ КРУГЛОЇ РІЗАЛЬНОЇ ПЛАСТИНКИ ІЗ ПОЗИТИВНИМ ЗАДНІМ КУТОМ | 62 |
| Ю.С. СОКОЛАН, К.А. ПАРШЕНКО АНАЛІЗ ЗМІН У НОРМУВАННІ ПРИРОДНОГО ОСВІТЛЕННЯ ПРИМІЩЕНЬ У ВІДПОВІДНОСТІ ІЗ ДЕРЖАВНИМИ БУДІВЕЛЬНИМИ НОРМАМИ | 67 |
| Н.О. КОСТЮК, А.І. ГОРДЕЄВ, О.А. ГОРДЕЄВ ДОСЛІДЖЕННЯ РОЗПОДІЛУ ЕНЕРГОВИТРАТ ПРИВОДУ ВІБРАЦІЙНОЇ МАШИНИ ПОРШНЕВОГО ТИПУ ДЛЯ ЗНЕЗАРАЖУВАННЯ ТА ЗМІНИ ВЛАСТИВОСТЕЙ ВОДИ | 73 |
| О.Г. ТИМОЩУК, С.П. ЛІСЕВИЧ ОЦІНКА ТОЧНОСТІ МЕТОДУ РОЗРАХУНКУ ЕФЕКТИВНОГО РІВНЯ ДЕФОРМАЦІЇ КОНТАКТУЮЧИХ ПОВЕРХОНЬ ЦИЛІНДРО-ПОРШНЕВИХ УЩІЛЬНЕНЬ | 79 |

О.М. ШИНКАРУК

Львівський Державний університет внутрішніх справ, м. Львів, Україна

О.М. ЯШИНА, О.Г. ОНИШКО

Хмельницького національного університету, м. Хмельницький, Україна

УПРАВЛІННЯ ЯКІСТЮ ПРОГРАМНИХ ВЕБ-СИСТЕМ ЗАСОБАМИ РОЗРОБКИ

У даній статті наведено результати досліджень існуючих засобів і визначення теоретичних аспектів застосування сучасних фреймворків для розробки програмного забезпечення, а також обґрунтування доцільності та впливу варіативності вибору на якість розроблюваних програмних додатків. Основними результатами дослідження є обґрунтування вибору фреймворків для розробки програмного забезпечення. Потрібно відмітити, що не зважаючи на велику кількість фреймворків, структурований аналіз здійснений досить незначною кількістю дослідників, що і обумовлює наукову новизну даного питання. Практична значущість полягає у можливості застосування тих чи інших фреймворків відповідно до потреб замовника та розробника для досягнення певних результатів при розробці програмного забезпечення.

Ключові слова. Фреймворк, CSS, програмне забезпечення, програмний продукт, мова програмування, управління якістю.

O. M. SHYNKARUK

Lviv State University of Internal Affairs

O. M. YASHYNA, O. H. ONYSHKO,

Khmelnyskiy National University, Khmelnytskyi, Ukraine

SOFTWARE QUALITY MANAGEMENT DEVELOPMENT TOOLS

At the present stage of development of software tools in the field of web development, the number of web frameworks (VFK) has increased sharply. All of them have a number of advantages and disadvantages, and therefore it is important to choose the right and appropriate framework or set for each project, because the right choice depends on the quality, implementation time and reliability of each software product being developed.

Generally speaking, web frameworks have definitely changed and improved their programming capabilities and become an integral part of the development process. There is a large amount of information related to VFK, but often this information contains only definitions and complex terms that do not give any idea about these tools. That is why the question of justifying the variable choice of frameworks and their impact on software quality is an urgent task at the present stage of programming development.

This article presents the results of research on existing tools and the definition of theoretical aspects of the use of modern frameworks for software development, as well as justification of the feasibility and impact of choice variability on the software applications quality. The main results of the study are the rationale for the frameworks' choice for software development. It should be noted that despite the large number of frameworks, the structured analysis was carried out by a rather small number of researchers, which determines the scientific novelty of this issue. The practical significance lies in the possibility of using certain frameworks in accordance with the needs of the customer and the developer to achieve certain results in software development.

Keywords. Framework, CSS, software, software product, programming language.

Постановка проблеми.

На сучасному етапі розвитку засобів програмування в галузі веб-розробки кількість веб-фреймворків (ВФК) різко зростає. Всі вони володіють певною низкою переваг та недоліків, а тому актуальним є питання вибору правильного та відповідного фреймворку або їх набору для кожного конкретного проекту, адже від правильного вибору залежить якість, час реалізації та надійність кожного розроблюваного програмного продукту.

Взагалі кажучи, веб-фреймворки однозначно змінили та покращили своєю появою можливості програмування і стали невід'ємною частиною процесу розробки. Існує велика кількість інформації, що стосується ВФК, але часто ця інформація містить тільки визначення та складні терміни, які не дають ніякого уявлення про дані засоби. Саме тому питання обґрунтування варіативного вибору фреймворків та їх вплив на якість програмного забезпечення є актуальним завданням на сучасному етапі розвитку програмування.

Аналіз останніх досліджень і публікацій.

Методологічною основою даного дослідження є праці таких дослідників як Jacek Schae, Маєрт О.Д., Масляк Т.А., Колесникова Т.А. та багато інших. Деякими загальними аспектами використання фреймворків займалися Татур Ю.Л., Білоконна К.В. Використання CSS-фреймворків у своїх роботах описують Масляк Т.А., Колесникова Т.А. Загалом. Разом із тим, в теперішній час, виконується велика кількість різноманітних досліджень, присвячених застосуванню тих чи інших ВФК та їх доцільності щодо розробки певного програмного продукту. Однак, потрібно відмітити, що проблема варіативності вибору веб-фреймворків при створенні програмних продуктів займає незначне місце у вітчизняних та зарубіжних роботах.

Не зважаючи на те, що дослідження в галузі застосування фреймворків постійно проводяться, наукових робіт, присвячених варіативності вибору та доцільності їх використання в тій чи іншій сфері розробки та впровадження програмного забезпечення недостатньо, що і обумовило виявлення до даного завдання підвищеного наукового та практичного інтересу.

Мета дослідження. Визначення теоретичних аспектів застосування сучасних фреймворків для розробки програмного забезпечення, а також обґрунтування доцільності та впливу варіативності вибору на якість розроблюваних програмних додатків.

Виклад матеріалу дослідження.

З урахуванням того, що веб-фреймворк є інструментом полегшення написання і запуску веб-додатків, розробнику не потрібно самостійно прописувати велику кількість коду та витратити час на пошук потенційних прорахунків і помилок [1,2,3]. В сучасних умовах існує можливість вибору з наявних мов для веб-програмування відповідного фреймворку, як для статичних, так і для динамічних сторінок. Так, в залежності від поставленого завдання, необхідним є вибір одного ВФК, що здатний задовільнити всі потреби, або поєднання декількох.

Слід приймати до уваги, що у ВФК є дві основні функції: робота на серверній стороні (бекенд) і робота на клієнтській стороні (фронтенд). При цьому, фронтенд-фреймворки пов'язані із зовнішньою частиною програми, тобто відповідають за інтерфейс, а бекенд-фреймворки відповідають за внутрішню частину додатку, тобто логікудодатку.

Як показує практика, серверні веб-фреймворки (СВФК) не дають можливості створення веб-додатків з насиченим інтерфейсом. Вони обмежені в своїй функціональності, однак забезпечують створення простих сторінок і різних форм. Також, СВФК здатні забезпечити формування вихідних даних і відповідати за безпеку в разі атак. Все це, безумовно, може спростити процес розробки. Серверні фреймворки, в основному, відповідають за окремі, але критично важливі частини програми, без яких неможливе стале функціонування. Найбільш поширеними фреймворками такого типу і мови програмування, з якими вони працюють, слід виділити наступні:

- Django – Python;
- Zend – PHP;
- Express.js – JavaScript;
- Ruby on Rails – Ruby.

На відміну від серверних, клієнтські веб-фреймворки (КВФК) ніяк не пов'язані з логікою програми. Цей тип фреймворків працює в браузері. З їх допомогою можна поліпшити і впровадити нові користувацькі інтерфейси. Фронтенд-фреймворки дозволяють створювати різні анімації і односторінкові додатки. Всі клієнтські фреймворки відрізняються по функціональності і використанню. Найбільш поширеними фреймворками такого типу слід виділити наступні:

- Backbone+Marionette;
- Angular;
- Ember.js;
- Vue.js.

Всі ці фреймворки використовують мову програмування - *JavaScript*.

Разом з типами фреймворків, що приведені вище, набули широкого використання для вирішення окремих завдань програмування й багатофункціональні фреймворки. *Meteor* відомий як фул-стек веб-фреймворк. Це означає, що він здатний задовільнити майже всі потреби як з боку клієнта, так і з боку сервера, що робить *Meteor* надзвичайно популярним. Його використання забезпечує заощадження часу, наприклад, на налагодження взаємодії між двома ВФК через *RESTAPI*. Використання, при цьому, ВФК *Meteor* забезпечить виконання відповідних процедур і прискорить процес розробки. Оскільки обидві сторони — серверна та клієнтська — працюють на одній мові, то існує можливість створення і використання для них одного коду. Одна з особливостей даного ВФК — «режим реального часу» — внесення змін в одному інтерфейсі, обумовлює зміни і в інших. Наприклад, при додаванні коментарів або зміні змісту документу або таблиці із загальним доступом, інші користувачі це також будуть бачити.

Окрему увагу слід приділити й розмірам (масштабам) фреймворків. Розрізняють ВФК, що здатні забезпечити рішення для всіх завдань, і більш легкі варіанти, які спеціалізуються на вирішенні конкретних завдань — такі фреймворки називаються мікрофреймворками. Останні не здатні забезпечити рішення всіх завдань, проте іноді краще розкласти функціональність на кілька підходів (фреймворки, мікрофреймворки, бібліотеки). При цьому, слід зауважити, що розширення функціональності мікрофреймворків можливе за допомогою сторонніх додатків і створенням невеликих проектів на їх основі, або шляхом поєднання мікрофреймворку з основним «великим» фреймворком.

Наприклад, якщо додаток заснований на ВФК *Django* і потрібні веб-сокети, то є доцільним використання мікрофреймворку *aiohhttp*. Або, якщо додаток не дуже великий і потрібна лише проста маршрутизація *URL* і шаблони з нескладним контекстом, то можливе використання ВФК *Flask* замість *Django*.

Незважаючи на те, що існуючі ВФК відрізняються один від одного, їх вибір може бути складним завданням, тому, слід навести кілька ознак, які є загальними для них усіх. Це архітектура і особливості, які важливі так, як і функції.

Архітектура всіх ВФК заснована на декомпозиції декількох окремих прошарків (додатки, модулі і інш.), що означає, можливість розширення функціональності в залежності від потреб і використання зміненої версії разом з кодом ВФК або використання сторонніх додатків. Така гнучкість вважається ще однією ключовою перевагою фреймворків. Існує безліч додатків з відкритим вихідним кодом (*open source*),

спільнот і комерційних організацій, які створюють програми або їх розширюють для популярних фреймворків, наприклад, *Django REST Framework*, *ng-bootstrap* і інші.

Таким чином, зміст кожного веб-фреймворка формується трома складовими: модель, представлення, контролер (model, view, controller), скорочено MVC, що подано на рис. 1.

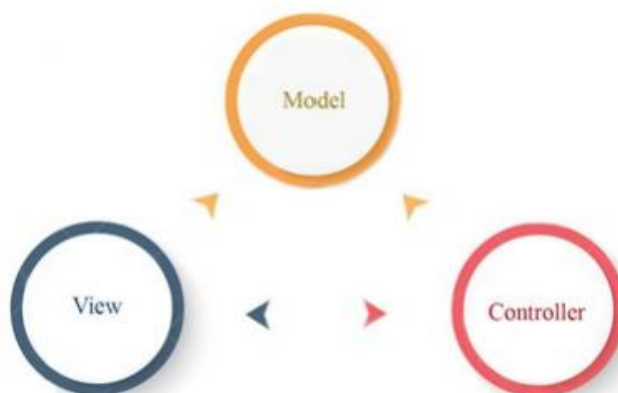


Рис. 1. Зміст веб-фреймворка.

Модель містить всі дані і рівні бізнес-логіки, її правила і функції.

Представлення відповідає за візуальне відображення даних, наприклад, діаграми, графіки і т. д.

Контролер просто трансформує дані для команд попередніх двох складових.

Вони невіддільні один від одного, тому важливо як слід у всьому розібратися, щоб уникнути помилок під час роботи програми.

Важливого значення набувають особливості фреймворків, основні з них подані на рис.2, які роблять їх багатofункціональними і зручними на практиці.



Рис. 2. Особливості веб-фреймворків.

Веб-кешування забезпечує збереження різних документів і дозволяє уникнути перевантаження сервера. Тому наявна можливість його використання в різних системах за певних умов. Також веб-кешування працює на серверній стороні. Наприклад, існують кешовані веб-сторінки на сторінці пошукової видачі *Google*.

Скафолдинг – це ще одна технологія, що полягає в автоматичному генеруванні типових частин програми або структури проекту. Це дозволяє істотно збільшити швидкість розробки і стандартизує кодову базу.

Система веб-шаблонів являє собою набір різних методологій і програмного забезпечення, реалізованих для створення і розгортання веб-сторінок. Для обробки веб-шаблонів використовуються шаблонізатори, що вважаються інструментом фреймворку, який відповідає за веб-публікацію.

Зіставлення *URL* – це функція фреймворків, що сприяє спрощенню доступу до сторінок веб-сайту, індексаційного пошуковими рушіями, створюючи привабливу назву.

Велика кількість типів веб-додатків підтримуються веб-фреймворками, що застосовуються, в основному, для створення таких додатків, як блоги, форуми, *CMS* і інші.

Наведена інформація щодо функціональності властива всім ВФК. Разом із тим, широкий асортимент фреймворків, як правило, ускладнює їх вибір. Тому, для полегшення розробки варто використовувати критерії, що обумовлюють кращий інструмент. Наприклад, ВФК *Angular* на мові програмування *JavaScript* – це фреймворк від *Google*, розроблений спеціально для створення динамічних веб-додатків, в тому числі інтерфейсних додатків без необхідності застосування інших плагінів або фреймворків. Його структура утримує низку цікавих функцій. Зокрема, використовуючи шаблони, відкривається можливість відображення інформації з моделі і контролера. При цьому, ВФК *Angular* підтримує архітектуру *MVC*, що відкриває можливість розділення додатку на *MVC* компоненти. При цьому, все інше управляється засобами фреймворка.

Всі ці функції є частиною платформи, яка сприяє розробці ефективного веб-сайту. Ось лише деякі приклади майданчиків, розроблених за допомогою цього фреймворка: *Netflix*, *Freelancer.com*, *GoodFilms* і інші.

Фреймворк *Ruby on Rails* створений мовою програмування *Ruby*. Однією з його переваг вважається докорінне спрощення і прискорення розробки веб-додатків за рахунок повторного використання коду. Такий підхід дозволяє додавати деякі додаткові функції. Серед популярних веб-сайтів, написаних на *Ruby on Rails*, можна виділити *Basecamp*, *Ask.fm*, *GitHub*, *500px* і інші. Але, основною перевагою цього фреймворку слід вважати швидку розробку з меншою кількістю написаного коду та помилок. До інших його переваг доцільно віднести гнучкість, що полягає в можливості широкого кола застосування (наприклад, від управління проектами до будівництва), швидкість, що полягає в здатності скорочення часу розробки веб-додатків приблизно на 30-40%, і можливість вношення змін в код. Саме тому фреймворк *RoR* ідеально підходить для довготривалих проектів.

Фреймворк *Yii* є платформою з відкритим вихідним кодом, що вбудована в *PHP5*. Завдяки зрозумілому дизайну і зручному інтерфейсу, цей фреймворк забезпечує надзвичайно просту і швидку веб-розробку. Він оптимізований по продуктивності, що дозволяє його використання для будь-яких проектів. Крім того, дана платформа утримує інструменти, які володіють здатністю налагодження і тестування додатку. Цікавою особливістю платформи слід вважати можливість увімкнення класів та об'єктів, лише, за потреби, що прискорює завантаження додатків. Дані функції сприяють створенню високопродуктивної структури, що сприяє розробці ефективних веб-сайтів (наприклад, *TACC*, *Craftcms*, *Hum Hub* і інші.).

Фреймворк *Meteor JS* розроблений на платформі *Node.js* дозволяє створювати різні *real-time* веб-додатки. Одна з основних особливостей *Meteor JS* — належна основа для створення простих сайтів особистого користування.

Meteor JS це ізоморфний *JavaScript* веб-фреймворк з відкритим вихідним кодом, що сприяє швидкому завантаженню сторінок. Крім того, завдяки інтегрованому стеку *JavaScript*, який простягається від бази даних кінцевого користувача до екрану, існує можливість виконання в 10 рядках коду те, що, зазвичай, розтягується на 1000 рядків.

Одна з особливостей даного фреймворку полягає в можливості використання одного коду при розробці під операційні системи *iOS*, *web*, *Android* або *desktop*. Також існує можливість використання різних популярних фреймворків та інструментів для створення функцій.

Фреймворк *Express.js* на платформі *Node.js* покриває цілий ряд важливих функцій плагінами, тому доцільним вважається використання для швидкої розробки веб-додатку і прикладного програмного інтерфейсу (*API*). Також даний фреймворк придатний до використання для створення мобільних додатків.

По суті, *Express.js* складається з *Angular* і бази даних *MongoDB*. Це означає, що для розробки веб-сайтів достатньо знання таких мов, як: *HTML*, *CSS* і *JavaScript*. А використовуючи модулі *npm*, відкривається можливість розширення функціоналу додатків. Також *Express.js* ідеальний для створення простих веб-сервісів.

Zend — це *opensource* фреймворк, що розроблений на мові *PHP*. Він орієнтований на розробку сучасних, надійних і безпечних веб-сервісів. При цьому застосовуються різні професійні пакети *PHP*, які роблять розробку веб-сайтів найвищого класу значно простіше і швидше. Крім того, ВФК використовує архітектуру *MVC*, що відокремлює базу даних і бізнес-логіку від представницького рівня. Це сприяє отриманню більш зрозумілого і чистого коду. Веб-фреймворк *Zend* базується на концепціях об'єктно-орієнтованого програмування, що відкриває можливість розширення різних компонент фреймворку. Також слід відзначити маршрутизацію, яка виконує свою роботу бездоганно, і функції кеша.

Django – найпопулярніший фреймворк для веб-розробки, написаний на мові програмування *Python* і використовує архітектуру *MVC*. Ключовими особливостями цієї платформи слід вважати наступні.

Швидкість. Головна мета фреймворка-допомогти розробникам зробити додаток, як можна швидше. Причому, на всіх етапах розробки — від ідеї до релізу. Ефективність і економічність – саме так можна сформулювати девіз *Django*.

Безпека. Як правило, деякі безпекові помилки, що пов'язані з «ін'єкціями» мови програмування структурованих запитів *SQL*, а також підркобою міжсайтових запитів і міжсайтовим скриптингом. *Django*

ефективно управляє наявними іменами користувача і паролями, а система аутентифікації користувача, як відомо, відіграє вирішальну роль.

Масштабованість. Більшість бізнес-сайтів використовують ВФК *Django* для швидкого задоволення потреб трафіку.

Багатофункціональність. Фреймворк включає в себе додаткові опції для допомоги з картами сайту, аутентифікацією користувачів, адмініструванням контенту, *RSS*-канали і багато іншого. Кожна з них надає істотну допомогу в процес веб-розробки.

ВФК *Laravel* вважається одним з кращих *PHP* фреймворків. Він здатний забезпечити легку розробку, як веб, так і мобільних додатків для невеликих сайтів і великого бізнесу. При цьому, ВФК *Laravel* відрізняється наявністю багатьох цікавих функцій, таких як, наприклад, техніка авторизації, об'єктно-орієнтовані бібліотеки, підтримка *MVC*, міграції баз даних, міжсайтові запити і інш.

До основних переваг цього фреймворку можна віднести:

- Можливість збільшення трафіку сайту, що обумовлює застосування до будь-якого браузера та пристрою.

- Гнучкість. Фреймворк має модульну структуру, що допомагає спростити сам веб-сайт і процес його розробки.

- *PHP* не потребує спеціальних способів обслуговування, що пов'язано з автоматичним завантаженням об'єкта, який міститься в програмному забезпеченні.

- Фреймворк *Laravel* може створювати унікальні *URL*-адреси, так як реалізує різні маршрути з однією і тією ж назвою.

Як показує досвід, кожен фреймворк володіє низкою переваг і недоліків, що проявляються при вирішенні певних задач. При виборі фреймворку необхідно враховувати декілька чинників, а саме:

- 1) можливості фреймворку повинні чітко відповідати завданням проекту. Зайвий функціонал не потрібен, але при цьому ніколи не повинно виникати обмежень при доопрацюванні проекту;
- 2) фреймворк повинен сприяти зменшенню часу розробки сайту;
- 3) для зручності використання фреймворку іншим розробником він повинен бути поширеним;
- 4) безпека фреймворка-один з найважливіших його показників.

У випадках прийняття рішення щодо вибору фреймворку, зазвичай порівнюється, з одного боку, заощаджений час за рахунок рішень, запропонованих фреймворком, а з другого боку, час, що витрачений на вивчення нової платформи. За таких умов вибір доцільно спрямовувати на технологію, що дозволяє в мінімальні терміни вирішити потрібне завдання. Разом із тим слід відмітити, що вибір до застосування нової технології може стати дерелом таких ризиків, як: внутрішні помилки, слабка підтримка і документація від розробника, неопрацьованість рішень під конкретні задачі.

Взагалі, при виборі найбільш ефективного ВФК доцільно проаналізувати завдання, чітко його формалізувати, вибрати «топ-5» відповідних варіантів, написати на них невеликий додаток і обрати найкращий.

Також, існує потреба оцінювання ступеню відповідності функціоналу, з переліку претендентів, вимогам проекту, а також тривалість застосування готового функціоналу. При цьому, обов'язково потрібно звернути увагу на можливість зворотньої сумісності, якщо фреймворк змінюється до версії *API*. За відсутності даної можливості такий ВФК використовувати не доцільно.

Як на думку авторів, обрання з відомого набору ВФК вважається нескладним завданням тому, що при проектуванні відразу видно потребу в тих чи інших технологіях. Далі постає завдання збільшення відомих фреймворків, тому, в даному випадку, варто, в першу чергу, вивчити фреймворки, які охоплюють різні технології і найбільш відрізняються один від одного. Наприклад, будь-який класичний *MVC*-фреймворк та ВФК іншої інфраструктури.

Також, слід відмітити, що вибір ВФК, як будь-якого інструменту, залежить від термінів проекту. Якщо проект короткотривалий, то слід обирати, наприклад, з групи *Bootstrap*, або щось нове і паралельно його вивчати. Якщо проект довготривалий, то краще формувати окрему бібліотеку структур.

Сучасні ВФК містять велику кількість готового функціоналу і помітно прискорюють розробку проектів. Тому, сьогодні більш актуальним вважається завдання вибору фреймворків, а не мов програмування. Зазвичай, вбудований готовий функціонал дозволяє зібрати досить ефективну бібліотеку структур. Важливою перевагою готового функціоналу, безперечно, залишається уникнення рутинного програмування. Разом із тим, одним з основних ризиків застосування різних ВФК і їх адаптації до конкретних задач, є «людський чинник». Як тільки завдання виходять за межі певного фреймворка або виникає необхідність у використанні іншого, часто виникає потреба залучення інших фахівців. Якщо сегментація технічних рішень поглиблюватиметься, то так і буде. Але в даний момент більшість фреймворків підтримують схожий функціонал, і сегментація не надто висока. Специфічні рішення є тільки для наймасовіших видів завдань.

Також, слід зауважити, що мови і фреймворки, особливо на пізніх стадіях розвитку, це ортогональні речі. У розвинених екосистемах фреймворки, як правило, мають *API* для роботи з декількома мовами і навпаки: мови з великою екосистемою мають низку фреймворків, готових для використання. В сучасних умовах, при старті проекту, вибір фреймворку, зазвичай, стоїть на першому місці, ще до вибору мови. Якщо ж, наприклад, дістався *legacy*-проект, то, як правило, вибір мови відсутній. Мікросервіси, один з головних

архітектурних трендів сьогодишнього дня, намагається вирішити в тому числі і цю проблему — «звільнити» розробників від використання конкретної мови або фреймворку, щоб кожна частина проекту розроблялася на найбільш адекватних для неї технологіях.

Слід виділити важливі чинники, що дають гарантію надійності і стабільності фреймворку. Гарантія надійності фреймворка – його відмовостійкість. Правильна розробка, оптимізація запитів, ефективне кешування – ось запорука успіху. Наявність стабільних релізів, велика кількість користувачів, наявність автоматизованих тестів, наявність активності, *opensource* (відкритий код).

Приймаючи до уваги активність застосування ВФК слід систематизувати проведені дослідження напрямів їх розвитку. Так, проведений аналіз свідчить про те, що фреймворки будуть рухатися по шляху «полегшення» розробки. Типові рішення, готові модулі та інше – все для того, щоб невідготівлена людина змогла розгорнути сайт.

Для багатьох популярних платформ період нарощування функціональності пройшов. На даний момент розвиток сфокусовано на збільшенні швидкодії. Це справедливо і для *frontend*-, і для *backend*-фреймворків. В якості часткового рішення розглядається поділ фреймворка на логічні модулі. Для проекту повинен збиратися комплект бібліотек, що вирішують тільки необхідні завдання. А одним з рішень слід вважати щільніше з'єднання клієнтської та серверної частин фреймворків з єдиним синтаксисом та логікою роботи із контентом. Принцип «спрощення», який описано вище, призводить до того, що фреймворки стають складнішими за змістом, але все простішими для користувача.

Розвиток фреймворків залежить від потреб ринку та можливостей браузерів і призначених для користувача девайсів. Подальше удосконалення фреймворків, безпосередньо, залежить від розвитку і появи нових платформ, як, наприклад, розробка технології *Canvas*, *WebGL*, *Node.js* – обумовила появу ВФК по роботі з ними.

Висновки. Отже, проаналізувавши ряд наукових робіт, що стосуються використання фреймворків для розробки програмного забезпечення наочним є висновок, що незважаючи на велику популярність даних засобів, існують й протиріччя у їх використанні.

Вибір відповідного фреймворка є однією з основних засад, що визначає якість майбутнього сайту. Кожну з технологій, що наведені у даній статті можна вважати вичерпною у всіх відношеннях тому, що їх використовують, як у практиці перетворення в графічний інтерфейс (*frontend*), так і для розробки варіанту архітектури програмного забезпечення (*backend*). Тому, слід обирати платформу у відповідності до зручностей щодо мов програмування.

Отже, *Agile* найбільше підходить для проектів із «відкритим кінцем», наприклад, запуск інтернет-сервісів, розробка комп'ютерних ігор, операційних систем. Однак, гнучкість може приводити до втрати фокусу та втрати передбачуваності. Дуже важливо відокремлювати помилки застосування гнучкого підходу від недоліків самої методології. Перш, ніж будуть реалізовані всі переваги підходу, потрібно буде деякий час на адаптацію до реалії конкретних завдань.

Література

1. Інтернет-аудиторія України. Статистика 2012-2013 и прогноз на 2014 год [Електронний ресурс] / Netpeak. Режим доступу ресурсу: [www/ URL: http://blog.netpeak.ua/internet-auditoriya-ukrainy-statistika-2012-2013-i-prognoz-na-2014-god](http://blog.netpeak.ua/internet-auditoriya-ukrainy-statistika-2012-2013-i-prognoz-na-2014-god).
2. Мейерт Д.О. Небольшая книга о HTML /CSS фреймворках [Текст] / Д.О. Мейерт// Орейли. – 2015. – 30 с.
3. Агалаков С.А. Статистические методы анализа данных [Текст] / С.А. Агалаков. – 18 с.
4. Білоконна К.В. Новітні технології розробки Web-додатків / Інноваційні та інформаційні технології в бізнесі та освіті [Електронний ресурс] : матеріали міжвузівського студентського вебінару (Вінниця, 21 жовт. 2015 р.) / відп. ред. Л.Б. Ліщинська. – Вінниця : ВТЕІ КНТЕУ, 2015. – С. 9-10. – Режим доступу: http://www.vtei.com.ua/doc/materialu_vebinary.pdf
5. WRITING ANGULAR 2 IN TYPESCRIPT [Електронний ресурс] — Режим доступу: <http://victorsavkin.com/post/12355572351/writing-angular-2-in-typescript> — Назва з екрана.
6. One framework. Mobile and desktop [Електронний ресурс]— Режим доступу: <https://angular.io/> – Назва з екрана.

References

1. Internet-auditorija Ukrainy. Statistika 2012-2013 i prognoz na 2014 god [Elektronnyj resurs] / Netpeak. Rezhim dostupak resursu: [www/ URL: http://blog.netpeak.ua/internet-auditoriya-ukrainy-statistika-2012-2013-i-prognoz-na-2014-god](http://blog.netpeak.ua/internet-auditoriya-ukrainy-statistika-2012-2013-i-prognoz-na-2014-god).
2. MejertD.O. Nebofshaja kniga o HTML /CSS frejmvorkah [Tekst] / D.O. Mejert// Orejli. – 2015. – 30 s.
3. Agalakov S.A. Statisticheskie metody analiza dannyh [Tekst] / S.A. Agalkov. – 18 s.
4. Bilokonna K.V. Novitni tehnologii rozrobki Web-dodatkov / Innovacijni ta informacijni tehnologii v biznesi ta osviti [Elektronnij resurs] : materialy mizhvuziv'skogo students'kogo vebinaru (Vinnicja, 21 zhovt. 2015 r.) / vidp. red. L.B. Lishhins'ka. – Vinnicja : VTEI KNTEU, 2015. – S. 9-10. – Rezhim dostupu: http://www.vtei.com.ua/doc/materialu_vebinary.pdf
5. WRITING ANGULAR 2 IN TYPESCRIPT [Elektronnij resurs] — Rezhim dostupu: <http://victorsavkin.com/post/12355572351/writing-angular-2-in-typescript> — Nazva z ekrana.
6. One framework. Mobile and desktop [Elektronnij resurs]— Rezhim dostupu: <https://angular.io/> – Nazva z ekrana.

Надійшла / Paper received : 21.11.2020 р. Надрукована/Printed :04.01.2021 р.

ДОДАТОК С
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

МЕТОД ТА ПРОГРАМНІ
ЗАСОБИ ПРЕПРОЦЕСИНГУ
ВХІНОГО ТЕКСТОВОГО
КОНТЕНТУ

Автор роботи:
ст.гр. ІПЗм-20-2 Онишко О.Г.
Керівник роботи:
к.т.н., доцент Яшина О.М.

Метою даного дослідження є підвищення якості методів та програмних засобів препроцесингу вхідного текстового коду за допомогою синтаксичного та семантичного аналізу текстів природною мовою шляхом розробки методу семантико-синтаксичного аналізу, в якому інтегруються ці два види аналізу. Дослідження в першу чергу спрямоване на створення методів аналізу текстів українською мовою.

Об'єкт дослідження - створення семантико-синтаксичних методів аналізу текстів українською мовою.

Предметом дослідження методи і алгоритми визначення рольових структур тверджень, а також методи і алгоритми семантичного розбору

НАУКОВА НОВИЗНА ОТРИМАНИХ РЕЗУЛЬТАТІВ

1. Удосконалено метод автоматичного визначення рольових структур висловлень заснованих на комунікативної граматики української мови.
2. Удосконалено метод комп'ютерного семантико-синтаксичного аналізу текстів, в який інтегровані методи побудови синтаксичних дерев залежності і визначення рольових структур висловлень. Застосування розробленого методу надає можливість покращити точність та повноту синтаксичного та семантичного аналізу.

ЗАВДАННЯ

1. Провести вивчення методів синтаксичного і семантичного аналізу текстів природною мовою.
2. Розробити метод визначення рольових структур тверджень в текстах українською мовою.
3. Розробити ефективний метод семантико синтаксичного аналізу, в якому інтегровані методи побудови синтаксичних дерев залежності і визначити рольові структури висловлень.
4. Реалізувати методи семантичного і семантико-синтаксичного аналізу. Розробити інтегровану систему семантико-синтаксичного аналізу.

5. Провести експериментальні дослідження методів семантичного та семантико-синтаксичного аналізу.
6. Розробити методи вирішення прикладних задач, які використовують результати семантичного і семантико-синтаксичного аналізу.
7. Провести експериментальне дослідження методів вирішення прикладних задач. Оцінити вплив розроблених методів семантичного та семантико-синтаксичного аналізу на якість вирішення цих задач.

Синтаксичний аналіз займає одне з найважливіших місць в ланцюжку обробки текстів природною мовою у багатьох програмних додатках. Він використовується в інформаційно-пошукових та інформаційно-аналітичних системах, в системах машинного перекладу та для вилучення інформації з текстів.

У сфері обробки текстів на природній мові, під синтаксичним аналізом мається на увазі порівняння лексем певної мови (природної або формальної) з її правописом.

Одиницею аналізу текстів на природній мові зазвичай є речення. З одного боку, синтаксичний аналіз є вирішенням проблеми розпізнавання

Він визначає, чи є речення граматично правильним з точки зору загальноприйнятих правил побудови висловлень у будь якій природній мові.

Наприклад, в англійській мові артикль ставиться перед словом, в українській мові, головне і залежне слово повинні бути однакові за родом, числом і відмінком. Це вимагається в системах автоматичної перевірки якості тексту в текстових редакторах.

Синтаксична структура може виступати або як проміжний результат, що є входом для семантичного аналізу, або як зручне представлення тексту на природній мові для вирішення високорівневих прикладних задач, наприклад в інформаційно-аналітичних системах або системах машинного перекладу.

Ідея інтеграції комп'ютерного синтаксичного і семантичного аналізу була висловлена багатьма дослідниками. На сьогоднішній день цілком очевидно, що через неоднозначність конструкцій в природній мові для побудови синтаксичних дерев з високою точністю неможливо обмежитися тільки знанням граматики мови у вузькому сенсі - тобто лінгвістичними знаннями про морфологію слів і про синтаксичні правила аналізу структур, які діють тільки з такими лінгвістичними категоріями, як частини мови і морфологічні признаки. Для того, щоб вирішити цю проблему, в загальному випадку потрібні статистичні знання про природню мову, знання семантики природньої мови.

Деякі дослідники пропонують альтернативи традиційному підходу, де розбір, в якому побудовано дерево синтаксису, строго передує семантичному аналізу, в якому формується семантичне представлення тексту, і розробляє підходи, які об'єднують ці два типи аналізу в одній процедурі. Аналіз тексту на природній мові, який виконується відповідно до цих підходів, назовемо семантико-синтаксичним

Семантичний-синтаксичний аналіз поєднує в собі підходи, які використовують як тісну інтеграцію синтаксичного, так і семантичного аналізу, наприклад, в єдиному алгоритмі, так і підходи, в яких вони розділені, але результати, отримані на двох етапах аналізу, впливають один на одного.

ЗАГАЛЬНИЙ АЛГОРИТМ РОБОТИ СЕМАНТИЧНОГО АНАЛІЗАТОРА

Алгоритм семантичного аналізу речення можна розділити на п'ять етапів:

- пошук предикатів слів та відповідних записів словника;
- пошук аргументів предикатів слів;
- призначати семантичні ролі знайденим аргументам відповідно до кожного знайденої словникової статті предикатів слів незалежно від інших аргументів;
- вибір найкращого розподілу семантичних ролей аргументами для даного словникової статті з урахуванням обмеження унікальності ролей шляхом вирішення задачі оптимізації завдань;
- вибір найкращої словникової статті предикатів слів та відповідного набору семантичних ролей.

ЩО ТАКЕ СЕМАНТИКО-СИНТАКСИЧНИЙ МЕТОД

Метод семантико-синтаксичного аналізу використовує інформацію, отриману в результаті семантичного аналізу, для корекції дерева синтаксису. Цей метод шукає слова, які потенційно можуть бути семантичними аргументами, але через помилки в дереві синтаксису речення вони не виявляються звичайним способом, і перевіряється можливість включення їх в рольову структуру предикатного слова.

Якщо рольова структура слова предикату дозволяє отримати додатковий аргумент, дерево синтаксису налаштовується відповідно до нової структури ролей. При коригуванні дерева виключаються потенційно виникаючі цикли, перевіряється проектність доданих зв'язків, а також здійснюється перевірка перетворення або за допомогою статистично-евристичного критерія, або за допомогою ансамблю бінарних класифікаторів на основі машинного навчання. На виправленому синтаксичному дереві по новому проводиться семантичний аналіз

З метою оцінки впливу розробленого методу на якість побудови синтаксичних дерев залежності семантико-синтаксична система аналізу порівнювалась з синтаксичним аналізатором, реалізованим за допомогою MaltParser. Він також використовувався для попереднього синтаксичного аналізу в системі семантико-синтаксичного аналізу. Тестування і навчання проводилися на корпусі ГРАК,

На текстовому підкорпусі проводилась оцінка якості встановлення синтаксичних зв'язків між предикатним словом і семантичними аргументами. Враховувалися тільки синтаксичні зв'язки, де основним словом є дієслово, дієприкметник, а залежним - є іменник, займенник, прийменник, якому підпорядкований іменник або займенник, питальне слово, а також зв'язки, де головне - іменник або займенник, а залежний - дієприкметник. Оцінчні метрики це були точність p , повнота r і F1-міра

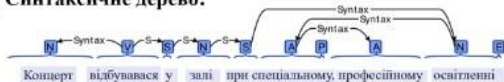
В експериментах порівнювали два способи перевірки виправлень в семантико-синтаксичному аналізі: за допомогою використання статистично евристичного критерію (K1), заснованого на перевірці частоти спільного виникнення ознак слів, і критерій(K2), заснований на ансамблі бінарних класифікаторів за допомогою машинного навчання. Таким чином, порівнювалися три аналізатори: аналізатор на основі MaltParser, K1 і K2

| Аналізатор | <i>p</i> , % | <i>r</i> , % | <i>F₁</i> , % |
|------------|--------------|--------------|--------------------------|
| MaltParser | 90,6 | 89,0 | 89,8 |
| K1 | 90,5 | 90,2 | 90,3 |
| K2 | 90,6 | 90,1 | 90,4 |

ОЦІНКА ЯКОСТІ
МЕТОДУ ВИЛУЧЕННЯ
АВТОРСЬКИХ ТЕРМІНІ З
ТЕКСТІВ НАУКОВИХ
ПУБЛІКАЦІЙ

Отримані результати показують, що розроблений метод семантико-синтаксичного аналізу дозволяє підвищити повноту встановлених синтаксичних зв'язків між предикатними словами і аргументами більш ніж на 1,1%, тим самим йому вдається встановити більше 10% всіх найскладніших синтаксичних зв'язків між предикатними словами і аргументами, які не були знайдені ГРАК. При цьому точність системи семантико-синтаксичного аналізу не змінюється.

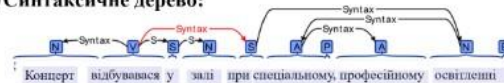
А) Синтаксичне дерево:



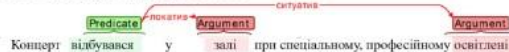
Рольова структура речення



Б) Синтаксичне дерево:



Рольова структура речення



ПРИКЛАД, В ЯКОМУ СЕМАНТИЧНА СИСТЕМА РОЗБОРУ ВИПРАВИЛА СИНТАКСИЧНЕ З'ЄДНАННЯ. НА РИСУНКУ А) – ПРИКЛАД РОБОТИ ГРАК, НА РИСУНКУ Б) – ПРИКЛАД РОБОТИ СИСТЕМИ СЕМАНТИКО-СИНТАКСИЧНОГО АНАЛІЗУ

ВИСНОВКИ

1. Розроблено новий метод автоматичного визначення рольових структур тверджень на основі комунікативної граматики української мови.

2. Розроблено новий метод комп'ютерного семантико-синтаксичного аналізу текстів, в якому інтегровані методи побудови синтаксичних дерев залежності і визначення рольових структур тверджень, що дає можливість підвищити точність і повноту синтаксичного і семантичного аналізу в порівнянні з реалізацією, в якій ці види аналізу виконуються окремо.

3. Розроблено та введено інтегровану систему семантико-синтаксичного аналізу. Система використовується для вилучення визначень та авторських термінів з текстів наукових публікацій

4. Експериментально показано, що використання рольової структури речення підвищує ефективність побудови правил вилучення визначень та термінів автора з текстів наукових публікацій.

СТАТТІ

- ▶ 1. T. Hovorushchenko , Ye. Hnatchuk , A. Herts, O. Onyshko. Intelligent Information Technology for Supporting the Medical Decision -Making Considering the Legal Basis. CEUR -WS. 2021. Vol. 2853. Pp. 72-82. - <https://www.scopus.com/authid/detail.uri?authorId=54420153900> (Q4)
- ▶ 2. О.Г.Онишко. Метод та програмні засоби препроцесингу вхідного текстового контенту.// Збірник наукових праць за матеріалами XII Всеукраїнської науково -практичної конференції « Актуальнв проблеми комп'ютерних наук АПК - 2021», 15-16 жовтня, Хмельницький: ХНУ 2021. – С.379-381
- ▶ 3. О.М. Шинкарук, О.М. Яшина, О.Г. Онишко. Управління якістю програмних веб-систем засобами розробки. / Вісник Хмельницького національного університету. – Технічні науки. – Хмельницький. - 2020.-№6. - С. 39- 45.

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.
здобувача вищої освіти
Онишко О.Г.
факультет ІТ, 2 курс, група ПЗм-20-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

23.11.2021 р.

дата


підпис



Имя пользователя:
Кафедра ИПЗ

Дата проверки:
05.12.2021 17:16:56 EET

Дата отчета:
05.12.2021 17:18:46 EET

ID проверки:
1009528660

Тип проверки:
Doc vs Internet + Library

ID пользователя:
100005589

Название файла: 5.2.12Онишко_mag

Количество страниц: 85 Количество слов: 17075 Количество символов: 136091 Размер файла: 2.44 MB ID файла: 1009539389

11.7% Совпадения

Наибольшее совпадение: 7.55% с Интернет-источником (<https://semanticua.blogspot.com>)

9.29% Источники из Интернета 197 Страница 87

2.9% Источники из Библиотеки 90 Страница 88

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 14

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Метод та програмні засоби препроцесингу вхідного текстового контенту»

Автор: Онишко Оксана Григорівна

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Яшина Оксана Миколаївна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 5 | Інше: | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті дипломної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, назвах розділів/підрозділів тощо та в назвах публікацій у переліку джерел посилання;

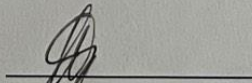
2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

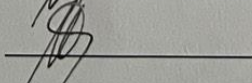
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 11,7% і адресується до 227 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломної роботи.

Керівник



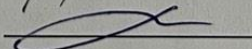
О.М. Яшина

Гарант ОП



О. М. Яшина

Завідувач кафедри



Л. П. Бедратюк

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Магістр__Онишко Оксана Григорівна_____

Тема_____Метод та програмні засоби препроцесингу вхідного текстового контенту

Спеціальність 121 – Інженерія програмного забезпечення**Обсяг дипломної роботи:**Кількість листів креслень 0 ; кількість сторінок записки 76

1.Короткий зміст ДР та прийнятих рішень У дипломній роботі удосконалено метод автоматичного визначення рольових структур висловлень, заснованих на комунікативній граматиці української мови. Удосконалено метод комп'ютерного семантико-синтаксичного аналізу текстів, в який інтегровані методи побудови синтаксичних дерев залежностей і визначення рольових структур висловлень. Застосування розробленого методу надає можливість покращити точність та повноту синтаксичного та семантичного аналізу.

На основі розроблених методів було створено програмний засіб препроцесингу вхідного текстового контенту.

2. Висновок про відповідність ДР поставленому завданню _____ Дипломна робота освітнього ступеня «магістр» у достатній мірі відповідає поставленому завданню як у теоретичній, так і в практичній її частині

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: _____

Розділ 1- Проведено огляд відомих методів синтаксичного та семантичного аналізу та визначено їх недоліки. Розділ 2 – Розроблено метод визначення рольових структур тверджень у текстах українською мовою та метод семантико-синтаксичного аналізу текстів. Розділ 3 – Проведенно дослідження розроблених методів семантико-синтаксичного аналізу тексту та визначення рольових структур тверджень. Розділ 4 – На основі розроблених методів створено програмний засіб препроцесингу вхідного текстового контенту

4.Позитивні сторони роботи _____ Застосування розробленого програмного засобу препроцесингу вхідного текстового контенту надає можливість підвищення якості вирішення задачі визначення рольових структур тверджень. Зокрема, надає можливість підвищити повноту на 1,7%, що призводить до збільшення F_1 -міри на 1,2%. При цьому точність не зменшується

5. Негативні сторони роботи Недостатньо детально в роботі прописана програмна реалізація

6. Оцінка графічного оформлення та пояснювальної записки роботи Оформлення роботи відповідає всім необхідним нормам та вимогам. Викладення матеріалу є логічним. Послідовним та стилістично грамотним

7. Відгук про роботу в цілому Кваліфікаційна робота магістранта Онішко Оксани Григорівни рекомендується до захисту.

8. Інші зауваження _____

9. Оцінка дипломної роботи Робота заслуговує на оцінку «добре»

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) _____

доц. кафедри Кваліфікаційної роботи
з інформатичних систем
Бабенко Р.В.

“ 6 ” 12 _____ 2021 р.


(підпис)