

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

Програмно-алгоритмічні засоби

семантико-синтаксичного аналізу законодавчих документів

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 123 “Комп’ютерна інженерія”  
(освітня програма - “Системне програмування”)

Науковий керівник роботи:

\_\_\_\_\_ (підпис)

доц. Щербакова М.Є.

\_\_\_\_\_ (ініціали, прізвище)

Консультант з юридичних питань:

\_\_\_\_\_ (підпис)

доц. Івчук Ю.Ю.

Консультант з юридичних питань:

\_\_\_\_\_ (підпис)

доц. Капліна Г.А.

\_\_\_\_\_ (ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_ (підпис)

Критська Я.О.

\_\_\_\_\_ (ініціали, прізвище)

Студент:

\_\_\_\_\_ (підпис)

Касьянов М.С.

\_\_\_\_\_ (ініціали, прізвище)

Група:

СП-16Дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень магістр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 123 "Системне програмування" (освітня програма - "Системне програмування")  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Касьянову Максиму Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмно-алгоритмічні засоби семантико-синтаксичного аналізу законодавчих документів

керівник проекту  
(роботи)

к.т.н., доц. Щербакова М.Є.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу

від " " \_\_\_\_\_ 2017 р. № \_\_\_\_\_

2. Строк подання студентом  
роботи \_\_\_\_\_

3. Вихідні дані до  
роботи \_\_\_\_\_

Навчальні програми дисциплін ВНЗ,  
Матеріали наукової практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналітичний огляд існуючих інструментів для аналізу

юридичних текстів. 2. Вибір технології для створення логічного аналізатора юридичних документів. 3. Семантико-синтаксичний

аналіз законодавчих документів. 4. Охорона праці та безпека в надзвичайних ситуаціях. Екологія.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових

креслень)

Електронна презентація

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Я.О. Критська		

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Огляд засобів симантико-синтаксичного аналізу	04.09.17 - 08.09.17	
2	Налаштування синтаксичного аналізатора SyntaxNet	11.09.17 – 21.09.17	
3	Формування технічного завдання на проектування	22.09.17 – 27.10.17	
4	Розробка і впровадження інформаційної системи синтаксичного аналізу	28.10.17 – 10.12.17	
5	Оформлення пояснювальної записки і графічного матеріалу	11.12.17 – 14.01.18	
6	Підготовка та подання магістерської роботи до захисту	15.01.18 - 19.01.18	

Студент

\_\_\_\_\_ ( підпис )

Касьянов М.С.

\_\_\_\_\_ (прізвище та ініціали)

Науковий керівник

\_\_\_\_\_ ( підпис )

Щербакова М.Є.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Касьянов М.С. Комплексна тема. Програмно-алгоритмічні засоби семантико-синтаксичного аналізу законодавчих документів.

Розглянуті існуючі програмно-алгоритмічні засоби, що використовуються для аналізу законодавчих та юридичних текстів. Запропонована теоретична модель глибокого семантико-синтаксичного аналізатора законодавчих текстів та доведена можливість його створення на практиці.

**Ключові слова:** СЕМАНТИЧНИЙ АНАЛІЗ, СИНТАКСИЧНИЙ АНАЛІЗ, ПРИРОДНА МОВА, ЮРИДИЧНИЙ ТЕКСТ, ЛОГІЧНИЙ АНАЛІЗ.

## АННОТАЦИЯ

Касьянов М.С. Комплексная тема. Программно-алгоритмические средства семантико-синтаксического анализа законодательных документов.

Рассмотрены существующие программно-алгоритмические средства, используемые для анализа законодательных и юридических текстов. Предложена теоретическая модель глубокого семантико-синтаксического анализатора законодательных текстов и доказана возможность его создания на практике.

**Ключевые слова:** СЕМАНТИЧЕСКИЙ АНАЛИЗ, СИНТАКСИЧЕСКИЙ АНАЛИЗ, ЕСТЕСТВЕННЫЙ ЯЗЫК, ЮРИДИЧЕСКИЙ ТЕКСТ, ЛОГИЧЕСКИЙ АНАЛИЗ.

## ABSTRACT

Kasyanov M.S. Complex issue. Software and algorithmic tools of lawmaking documents semantic-syntactic analysis.

Existing software and algorithmic tools used for the analysis of legislative and legal texts are considered. A theoretical model of the deep semantic-syntactic analyzer of the legislative texts is proposed and proved possibility of its creation in practice is.

**Key words:** SEMANTIC ANALYSIS, SYNTACTIC ANALYSIS, NATURAL LANGUAGE, LEGAL TEXT, LOGICAL ANALYSIS.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	6
ВСТУП	7
1. АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ ІНСТРУМЕНТІВ ДЛЯ АНАЛІЗУ ЮРИДИЧНИХ ТЕКСТІВ	10
1.1 Сучасні NLP інструменти	11
1.1.1 Stanford parser	14
1.1.1.1 Автоматична анотація семантичних об'єктів і ролей	14
1.1.1.2 Пошук і класифікація іменованих об'єктів в юридичних документах	14
1.1.1.3 Huang	15
1.1.1.4 Wyner	15
1.1.2 Apache OpenNLP	16
1.1.2.1 Intelligent Amplification for Cybercrime (IAC)	16
1.1.2.2 Vico-Calegari	16
1.1.2.3 Schilder	16
1.1.3 Spacy и TensorFlow	17
1.1.3.1 Регулировка и воссоздание EuroVoc с правовыми таксономиями	18
1.1.3.2 Schrading	19
1.1.4 Gensim	19
1.1.4.1 Автоматичне виявлення трансферів директив Європейського союзу	19
1.1.5 Boxer/CGG parser	20
1.1.5.1 Витяг деонтичних правил з технічних документів	21
1.1.5.2 Wyner	21
1.1.6 JFLEX	22
1.1.6.1 VO-ECLI	22
1.1.7 Parse-IT	22
1.1.7.1 PermitME	22
1.1.7.2 Regorous	23
1.1.8 TULE parser	23
1.1.8.1 ProLeMAS	23
1.1.8.2 DAPRECO	24
1.1.8.3 Eunomos	24
1.1.8.4 OpenSentenze	24
1.1.9 SPeLT	24
1.1.9.1 EuCases	25
1.1.9.2 Проект Швейцарської канцелярії	25
1.1.9.3 Проект верховного касаційного суду	25
1.1.9.4 Проект FAO	25
1.2 Результати	26
1.3 Завдання проекту	28
2 ВИБІР ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ЛОГІЧНОГО АНАЛІЗАТОРА ЮРИДИЧНИХ ДОКУМЕНТІВ	31
2.1 Супутні роботи	34
2.2 Логічна схема Хоббса	35
2.3 Обробка деонтичних обґрунтованих аргументів у правовій інтерпретації	37
2.4 Логіка Вводу/Виводу (I/O)	39
2.5 Об'єднання логіки Вводу/Виводу та логіки Хоббса	41
2.6 Узагальнення аксіом логіки вводу/виводу	42
2.7 Формалізація прикладів	43
2.8 Глибокий семантико-синтаксичний аналізатор	44

3 СЕМАНТИКО-СИНТАКСИЧНИЙ АНАЛІЗ ЗАКОНОДАВЧИХ ДОКУМЕНТІВ	45
3.1 Синтаксичний аналіз законодавчого тексту.	45
3.1.1 Налаштування необхідних технічних та програмних засобів для проведення синтаксичного аналізу.	45
3.1.1.1 Комп'ютерна техніка	45
3.1.1.2 Програмні засоби	45
3.1.2 Структура бази даних та інтерфейс користувача.	46
3.1.3 Синтаксичний парсер	48
3.1.4 Проблематика аналізу текстів різноманітних мов	49
3.1.5 Аналіз результатів синтаксичного розбору тексту.	53
3.2 Семантичний аналіз	55
3.2.1 Виявлення суб'єктів та об'єктів правовідносин	56
3.3 Висновки до розділу	57
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ	59
4.1 Загальні питання з охорони праці	59
4.1.1 Правові та організаційні основи охорони праці	59
4.1.2 Організаційно-технічні заходи з безпеки праці	61
4.2 Аналіз стану умов праці	62
4.2.1 Вимоги до приміщень	62
4.2.3 Навантаження та напруженість процесу праці	65
4.3 Виробнича санітарія	65
4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	66
4.3.2 Пожежна безпека	66
4.3.3 Електробезпека	69
4.4 Гігієнічні вимоги до параметрів виробничого середовища	70
4.4.1 Мікроклімат	70
4.4.2 Освітлення	70
4.4.3 Шум та вібрація, електромагнітне випромінювання	73
4.4.4 Вентилювання	73
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	74
4.6 Охорона навколишнього природного середовища	78
4.6.1 Загальні дані з охорони навколишнього природного середовища	78
4.6.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі	79
4.6.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі	80
4.7 Висновки до розділу	85
ВИСНОВКИ	86
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	87
Додаток А	98
Додаток Б	123

**ПЕРЕЛІК СКОРОЧЕНЬ**

- СУБД – система управління базами даних
- NLP – обробка природної мови (Natural Language Processing)
- ГССА – глибокий семантико-синтаксичний аналізатор
- NL – природна мова (Natural Language)
- ССГ – комбінаторний алгоритм категоріальні граматики
- DRS – структура подання дискурсу
- ЄС – Європейський Союз
- POS – частини мови (Part of Speech)
- NER – іменоване розпізнавання сутностей (Named Entity Recognition)
- TS – сегментації тексту (Text Segmentation)
- LSA – латентний семантичний аналіз
- LDA – латентне розміщення Діріхле (Latent Dirichlet Allocation)
- LQA – юридичне Питання-Відповідь (Legal Question Answering)
- QA – Запитання-Відповідь (Question Answering)
- RNN – Рекурентні нейронні мережі
- pLSA – ймовірнісний латентний семантичний аналіз
- NLTK – бібліотека природної мови (Natural Language Toolkit)
- ЄК – Європейська комісія
- NIM – національні заходи по здійсненню
- SVD – сингулярна декомпозиція значення
- DFA – детерміновані кінцеви автомати
- DL – деонтична логіка (Deontic Logic)
- NLS – семантика природної мови
- FOL – логіка першого порядку
- I/O – логіка Вводу/Виводу
- LLD – мова Маккарті для юридичного дискурсу
- DRT – теорія подання дискурсу
- MRS – семантика мінімальної рекурсії ()
- BPC – відповідність бізнес-процесів

## ВСТУП

**Актуальність теми.** Вчені-юристи та юристи-практики відчують себе все більш і більш розгубленими перед розширеним набором законів і прецедентного права, які в умовах глобалізації сучасного світу набувають більш міжнародний характер.

Наприклад, європейське законодавство оцінюється в 170 000 сторінок, з яких більше 100000 сторінок були випущені за останні десять років. Крім того, законодавство доступно в основному в неструктурованих форматах, що ускладнює користування цими даними. Оскільки законодавство стає більш складним, суперечливим і мінливим, необхідні більш складні методології для аналізу, представлення та обґрунтування правових знань.

Управління великими сховищами правових норм і правил, а також семантичний доступ та обґрунтування цих норм є ключовими проблемами у правової інформатики, яка переживає зростання активності вже на промисловому рівні. Зокрема, необхідно вирішувати як концептуальні завдання, такі як роль юридичної інтерпретації і осмислення даних, так і обчислювальні проблеми, такі як обробка великих об'ємів юридичних даних та складність дотримання нормативних вимог.

Якщо уважно подивитися на системи правової інформатики, описані в першому розділі, то можна помітити, що в основному розробники отримують юридичну інформацію з юридичних документів з метою вдосконалення систем пошуку, з метою зручності вилучення та зберігання юридичної інформації в інформаційних базах, тобто створюються інструменти, що допомагають в роботі кваліфікованих юристів.

Хоча ці системи допомагають орієнтуватися в законодавстві і отримувати інформацію, їх загальна корисність обмежена через те, що основна увага приділяється термінологічним питанням, не звертаючи уваги на семантичні аспекти, які допускають юридичні міркування. Але саме семантичні аспекти, що допускають юридичні міркування, і створюють правила, які використовуються для керівництва проведенням соціальної взаємодії. Ці правила вказують, які дії заборонені, які можливі і які обов'язкові.

Очевидне завдання як раз і полягає в отриманні таких правил з текстів юридичних документів. Для того, щоб зробити це, ми повинні усунути розрив між джерелами природної мови (NL) і машиночитаним поданням правил.

Усунення такого розриву підвищить ефективність аналізу законодавчих текстів законодавства на слов'янських мовах, дозволить створити інструменти, які можуть допомогти створювати прозоре законодавство, допоможуть в пошуку протиріч у законодавчих нормах, дозволять уникати матеріальних втрат у зв'язку з правовою неосвіченістю та протидіяти корупції в органах судової та державної влади.



Таким чином розробка є актуальною.

**Мета і задачі дослідження.** Метою магістерської роботи є підвищення ефективності аналізу законодавчих текстів законодавства на слов'янських мовах, за рахунок створення інформаційної технології для смислового аналізу законодавчих текстів українського законодавства, за допомогою існуючих програмно-алгоритмічних засобів семантико-синтаксичного аналізу та доказ можливість його практичного використання.

Для досягнення поставленої мети в роботі вирішуються такі **задачі**:

1. Аналітичний огляд існуючих інструментів для аналізу юридичних текстів.
2. Вибір технології для створення логічного аналізатора юридичних документів
3. Семантико-синтаксичний аналіз законодавчих документів.
4. Охорона праці та безпека в надзвичайних ситуаціях. Екологія.

*Об'єкт дослідження* – Процес розуміння юридичного сенсу документів українського законодавства викладеного слов'янськими мовами.

*Предмет дослідження* – засоби та алгоритми семантико-синтаксичного і логічного аналізу законодавчих текстів.

**Методи дослідження.** При вирішенні загальної та часткових наукових задач використовувалися рекурсивні алгоритми, логічні алгоритми, такі як, логіка вводу/виводу, логічна схема Хоббса, логіка першого порядку. При розробці використана система Odoo. Мова розробки сервера - Python v. 2.7. Для роботи web-клієнта використовується Node.js. Сервер СУБД Postgresql 9.5. Синтаксичний парсер - SyntaxNet, який створений на основі штучної нейронної мережі TensorFlow.

**Наукова новизна одержаних результатів.** Дістала подальшого розвитку моделі глибокого семантико-синтаксичного аналізу, що використовується для слов'янських мов базується на існуючих інструментах семантико-синтаксичного і логічного аналізу законодавчих текстів.

**Практичне значення одержаних результатів** полягає у тому, що на підставі проведених досліджень і запропонованих методів:

- запропонований глибокий семантико-синтаксичного аналізатор, за умови його створення, може бути використаний в правовій та багатьох інших напрямках інформатики;
- розроблено браузерне програмне забезпечення, що дозволяє швидко проводити синтаксичний аналіз текстів NL;

**Зв'язок роботи з науковими програмами, планами, темами.** Магістерська робота виконувалася протягом 2016-2018 рр. згідно з планами науково-дослідних робіт кафедри комп'ютерної інженерії в межах НДР «Дослідження у галузі вищої освіти і практики інституційної співпраці» (№ ДР 0113U002236).

**Особистий внесок здобувача.** Усі основні положення, результати, висновки та рекомендації магістерської роботи отримано автором самостійно.

**Апробація результатів магістерської роботи.** Попередні результати досліджень оприлюднено на II регіональному форумі «ІТ-Ідея 2016».

**Публікації.** За темою магістерської роботи з викладенням її основних результатів опубліковано тези доповіді.

**Структура та обсяг магістерської роботи.** Магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Робота викладена на 133 сторінках машинописного тексту, містить 87 сторінок основного тексту, 8 рисунків, 9 таблиць, 2 додатки на 35 сторінках. Бібліографічний список включає 157 найменувань.

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ ІНСТРУМЕНТІВ ДЛЯ АНАЛІЗУ ЮРИДИЧНИХ ТЕКСТІВ І ЗАВДАННЯ РОБОТИ

В цьому розділі наведено та стисло описані основні інструменти NLP, більшість з яких дійсно є інструментами багатоцільовими, тобто конкретно не використовується для обробки тільки юридичних текстів, а також конкретні проекти та науково-дослідної діяльності правової інформатики, в яких вони були використані. Також в цьому розділі формалізовані завдання цієї магістратської роботи.

Матеріали цього підрозділу ґрунтуються на роботі[1] "D2.1 Collection of state-of-the-art NLP tools for processing of legal text" Contributors: UL, UNITO, UNIBO, INRIA, DATA61 (2017)

#### 1.1 Сучасні NLP інструменти

Сучасні підходи в NLP чітко показали, що обробка в значній мірі залежить від області. Внутрішньообласні рішення більш стабільні, ніж позаобласні. Юридичні тексти, зокрема, використовують конкретні лексики і вирази природної мови, які рідко зустрічаються в юридичних текстах. Доступні інструменти NLP в основному, залежать від мови: в той час як є основні інструменти NLP (pos-tagger, parsers і т. д.), є з хорошою підтримкою англійською мови, так як це сама поширена в світі мова, для інших мов, наприклад, російською, навіть якщо вони досить поширені, обмежена доступність.

Методи NLP можуть бути класифіковані у вигляді двох основних класів:

- Статистичні методи, використовувані для пошуку загальної інформації в документах. Наприклад, для зіставлення документа, статті, абзацу і т. д. з певними мітками (з кінцевого набору доступною мітки), яка описує тему, про яку йде мова (класифікація).
- Методи, засновані на правилах, які використовуються для пошуку конкретної інформації в документах. Наприклад, для здобуття посилань на юридичні документи, дати, місця, імена осіб і т. д., а також деякі релевантні концепції (зв'язування об'єктів).

У літературі запропоновано правила на основі методик, що використовуються для класифікації чи, просто пошуку загальної інформації. Приклад [2], де на основі процедурних правил була запропонована класифікація модифікаторів станів в юридичних текстах, наприклад, скасування, заміни і т. д.

Статистичні методи також прийняті в разі використання конкретних сценаріїв отримання інформації, і вони засновані на заданні статистично значущих моделей, які спрямовані на вираження конкретних лексичних утворень. Наприклад, робота [3] уписує методи цього типу.

Нарешті, звичайно можна розробити гібридний підхід, тобто статистичні підходи, які виводять ряд правил, які можуть бути вручну налаштовані пізніше. Цей напрямок досліджень досліджується в TULE parser [4].

Добре відомо, що статистичні базовий методи базуються на правилах "trade-off".

Часто, статистичним інструментам NLP потрібен початковий корпус анотованих даних, на основі яких можливо бути навчати модель, яку NLP в свою чергу буде використовувати як інструмент, щоб анотувати нові тексти.

За умови що такі корпуси доступні, підготовка статистичної моделі, здатної обробляти нові текст - відносно швидкий етап. Однак, часто буває, що статистичні моделі, все ще навчаючись на великих корпусах, здатні правильно аналізувати нові мовні конструкції, але, в той же час, вже не в змозі правильно проаналізувати лінгвістичні конструкції які були раніше оброблені. Результати, точність цих аналізаторів має тенденцію коливатися навколо певного максимального значення точності (як правило, 80-90% точності, залежно від областей даних, обсягу даних, технічні параметрів і алгоритмів).

В інших сценаріях статистичні підходи можуть бути неконтрольованими. У цих випадках математичні моделі націлені на отримання шаблонів, які відображають певну статистичну значущість і специфіку контексту. Ці шаблони можуть містити в собі лінгвістичні функції, а також числові частоти, і їх можна використовувати для зіставлення (навіть частково) з новими вхідними даними. Цей механізм дозволяє структурувати неструктуровані дані автоматично, у вигляді кластерів, які поділяють рівні внутрішнього подібності.

З іншого боку, з допомогою правил, коли знайдені нові лінгвістичні конструкції, досить додати правила для їх покриття. І, оцінюючи засновану на правилах систему відносно корпусів, в принципі можна досягти 100% точності, оскільки після додавання нового правила легко автоматично визначити, які лінгвістичні конструкції більше не обробляються так потрібно, і яке правило має бути відповідним чином змінено. Основним недоліком інструментів, заснованих на правилах, є те, що створення відповідного набору правил є дуже повільним і трудомістким процесом, оскільки в більшості випадків вони виставляються вручну.

Як було сказано вище, базові інструменти NLP зазвичай є багатоцільовими, тобто визначені для обробки природної мови текст у цілому. Іншими словами, доступні інструменти NLP не розроблені для обробки юридичного тексту.

Звичайно, щоб мати хороші результати, інструменти NLP необхідно навчати і налаштовувати на юридичних текстах, з тим щоб вловити особливості юридичної мови.

Інструменти NLP зазвичай класифікуються таким чином, хоча можна знайти інструменти, які поєднують два або більше з них.

Інструменти NLP зазвичай класифікуються наступним чином (хоча можна знайти інструменти, які поєднують два або більше з них):

**Part of Speech (POS, Частини мови) теги:** інструменти NLP, які, взявши вільний текст у вхідних даних, класифікують кожне слово у відповідності зі своєю частиною мови. Доступні теги POS розрізняють зміст частин мови (іменники, дієслова, прикметники, прислівники) і функціональну частину мови (статті, прийменники, розділові знаки і т. д.). Крім частини мови, багато POS теги пов'язують інформацію про перегини слів (стать, число, настрій тощо).

**Named Entity Recognition (NER, Іменоване розпізнавання сутностей):** завдання пошуку і класифікації названих об'єктів у текстах у відповідності з наперед визначеним набором категорій. У більшості випадків це категорії власних іменників, підкатегоризовані у власних іменах людей, міст, названих артефактах і т. д. Крім власних іменників, інструменти NER здатні розпізнавати вираження часу, кількості, грошові цінності тощо, а також конкретні поняття або лінгвістичних моделей.

**Синтаксичні аналізатори:** інструменти NLP, які беруть результати POS тегів у вільному тексті, та забезпечують опис того, як слова організують речення. Існує два підходу: підхід оточення, тобто групування суміжних слів у фрази і залежний підхід, тобто зв'язування слів у відповідності з визначеним набором граматичних категорій (суб'єкт, об'єкт тощо).

**Сегментація тексту:** Метою сегментації тексту (Text Segmentation, TS) є визначення меж тематичного зрушення в документі. Дослідження структури дискурсу показали, що документ зазвичай є суміш тем і підтем. Зміна може бути помічена із змінами в моделях використання лексики. Процес ділення тексту на частини різних актуальних для тексту тем називаються сегментація тексту (Text Segmentation). Текстові одиниці (речення або параграфи), що складають сегмент, повинні бути узгодженими, тобто демонструвати сильні граматичні, лексичні та семантичні єдності. Додатки TS включають в себе Information Retrieval (IR, відновлення інформації), уривчасте відновлення та узагальнення документа.

**Тематичне моделювання:** Тематичні моделі є фундаментальними інструментами для здобуття закономірностей і шаблонів, що забезпечують автоматичні способи організації, пошуку і розуміння великих колекцій даних. Загальне базове припущення полягає в тому, що документи мають латентну семантичну структуру, яка може бути виведена з розподілу слів і документів. Латентний семантичний аналіз (LSA) [5] - метод на основі лінійної алгебри, який розташовує слово-матрицю збігів слів документа у скороченому просторі, так що слова, близькі в новому просторі, розглядаються як схожі. Його ймовірнісна і генеруюча версія (pLSA) [6] додає латентну змінну контексту для кожного слова, яка явно враховує багатозначність. Латентне розміщення Діріхле (Latent Dirichlet Allocation, LDA) [7] є повністю байєсівської ймовірнісною версією LSA. Враховуючи склад документів, ідея, що лежить в основі LDA,

полягає в тому, що всі документи мають один і той же тематичний набір, але кожен документ демонструє цю тематику в різних пропорціях в залежності від слів, які присутні в цьому документі. Теми, в свою чергу, визначаються як різні розподілу ймовірностей за словами фіксованого словника, але вони інтерпретуються шляхом обмеження уваги до слів з найвищою оціненою частотою. Спостерігаються тільки документи, в той час як тематики, однодокументний розподіл за тематикою і однокомпонентне однослівну тематичне призначення є латентними структурами, отриманими з даних.

**Keyphrase Extraction (здобуття ключової фрази):** Необхідність аналізу текстів і здобуття суттєвих ознак, які описують загальний контент і семантику тексту, - це виклик, який знайшов застосування в безлічі додатків при обробці природної мови. Ключові фрази, іноді звані ключовими словами, являють собою концепції, які забезпечують короткий опис вмісту документа і це показує, що вони корисні при індексації документів, наприклад, використовуються бібліотекарями при категоризації книг, кластеризації та візуалізації документів, узагальнення тексту, категоризації тексту, текстової анотації та тематичної індексації та для виявлення нових тематик.

**Семантична анотація:** тексти, а також юридичні тексти в цієї області проекту можуть бути пов'язані з семантичною інформацією або тегами, які забезпечують семантичний доступ, пошук, управління та аналіз великих колекцій даних. Ці теги можуть бути навіть структуровані якимось чином, чи вони можуть відображати онтологічну організацію понять. Тим не менш, зв'язок з тегами може відбуватися на різних рівнях деталізації. У світлі цього ми визначаємо дві специфікації:

- Семантичні ролі - це коли короткі фрагменти текстів відіграють семантичну роль щодо описаній ситуації. Це схоже на збіг з іменованими об'єктами, але навіть названі суб'єкти можуть грати різні ролі в різних реченнях, тому семантичне зерно в цьому випадку набагато глибше.
- Семантичні категорії (або класи) - це, як правило, ярлик для опису асоціації середньострокових текстів з тегами, що представляють теми або типи документів. Семантичним категоріям часто потрібно машинне навчання для створення моделей, здатних автоматично класифікувати вхідні тексти в одній або кількох категоріях, ґрунтуючись на тому, що вони дізналися з позначених прикладів (наборів для навчання).

**Питання-Відповідь:** в обмеженій області проекту MIREL, Юридичне Питання-Відповідь (Legal Question Answering, LQA) - це завдання що відповідає на питання задане натуральною мовою щодо юридичного тексту. Зокрема, ми ставимо конкретне завдання що вимагає відповіді типу «ТАК» або «НІ», де «ТАК» означає, що питання пов'язане з текстом (та «НІ» в іншому випадку).

### **1.1.1 Stanford parser**

Бібліотека Stanford NLP є одним з найбільш використовуваних інструментів NLP. Він забезпечує набір інструментів розбору природної мови, які охоплюють більшість методів, описаних в попередньому розділі цього огляду. Він видає основу форми слів, частин речі, найменовані сутності і нормалізує токени, що представляють дати, час і числові величини. Потім він знаходить, як терміни фраз пов'язані через синтаксичні залежностей.

Структура Stanford NLP заснована на статистичній моделі, і вона відома як надійна і швидка навіть на великих вхідних даних. Він підтримує кілька мов, відмінних від англійської, і може бути запущений як простий веб-сервіс. Це один з найбільш використовуваних парсерів в світі.

#### **1.1.1.1 Автоматична анотація семантичних об'єктів і ролей**

UNITO вже зіткнувся з проблемою автоматичного отримання структурованої інформації при спробі поліпшити семантичний пошук і створення онтологій в текстових базах даних. UNITO запропонував і реалізувати підхід, який в першу чергу покладається на Стенфордський парсер для перетворення тегів POS і синтаксичних залежностей в узагальнені функції, спрямовані на захоплення цільових семантичних одиниць навколо лінгвістичної мінливості.

Ці нові визнані дані подаються в класифікатор векторної машини підтримки, яка обчислює модель покликану автоматизувати семантичну анотацію. Система може витягувати семантичні сутності і ролі всередині вмісту юридичних текстів [3].

#### **1.1.1.2 Пошук і класифікація іменованих об'єктів в юридичних документах**

INRIA разом з партнером Cordoba використав частину бібліотек Stanford NLP для їх роботи що до популяції онтології юридичних онтологій. Ідея полягає в тому, щоб використовувати Вікіпедію як джерело ручних анотованих прикладів юридичних осіб. Вони вирівнюють YAGO, онтологію засновану на Вікіпедії, і LKIF, онтологія, спеціально розроблена для юридичної області. Завдяки цьому вирівнюванню вони можуть ефективно заповнювати онтологію LKIF, щоб отримати приклади для навчання Named Entity Recognizer(NER) і класифікатор, які будуть використовуватися для пошуку і класифікації об'єктів в юридичних текстах.

Вони отримали оцінку своєму підходу, навчаючи лінійний класифікатор, а саме машину Support Vector Machine (SVM) з лінійним ядром і модель Stanford CRFClassifier для NER. В якості додаткової базової лінії для оцінки вони отримали результати роботи системи Stanford NER, навчаючи її своїм корпусом анотаціями Вікіпедії для класів LKIF.

#### **1.1.1.4 Huang**

Число документів, що застосовуються в глобальному патентування, швидко росте в останні роки. Ця система спрямована на категоризацію патентних документів з використанням Stanford NLP разом з теорією грубих множин. Причиною створення цієї системи стало те, що для класифікації патентної документації, ручна перевірка не тільки повільна, але і дорога. Це дослідження спрямоване на пошук ефективного підхід для автоматичної класифікації патентних документів.

[8] використовувала базу даних WIPO-alpha від Всесвітньої організації інтелектуальної власності для двох експериментів, перший з яких об'єднує Stanford Parser і Rough Set Theory, щоб побудувати N-грам алгоритм витяга предковичної фрази. Другий експеримент, спрямований на об'єднання стоп-слів і Теорії грубих множин. Результати показали, що середня F-оцінка запропонованого методу 98.5%, що вище традиційних методів.

#### **1.1.1.4 Wyner**

У цій роботі [9] автори зіткнулися з проблемою виявлення, витяга і формалізації умовних і нормативних правил з правових норм. Наприклад, існують юридичні матеріали у вигляді XML, що робить їх доступними для пошуку та побудови зв'язків в Інтернеті, наприклад, такими як SEN MetaLex, поряд з національними стандартами, наприклад, у Сполучених Штатах.

Проте ідентифікація, витяг і формалізація правил і раніше являє собою дуже важливу і трудомістку задачу, створює суттєве вузьке місце між семантичним змістом вихідного матеріалу, вираженим природною мовою, і автоматичним використанням цього вмісту на основі комп'ютерної обробки. Щоб усунути вузьке місце, застосовувалися методи обробки природної мови (NLP).

Основний внесок [9] - це ідентифікація та витяг високорівневих компонентів правил з регуляторних актів на англійській мові, застосовуючи поширені, широко доступні сучасні інструменти NLP, такі як Stanford Parser. Потім автори надали модульну структуру з відкритим вихідним кодом, а також прозору методологію і матеріали з відкритим вихідним кодом.



### **1.1.2 Apache OpenNLP**

Бібліотека Apache OpenNLP являє собою набір інструментальних засобів для машинного навчання для обробки тексту на природній мові. Він підтримує найбільш поширені завдання NLP, такі як токенизація, сегментація речення, теги частини мови, витяг іменованого об'єкта, фрагментація, розбір і узгодження посилань. Ці завдання зазвичай необхідні для створення більш досконалих служб обробки тексту.

OpenNLP також включає в себе максимальну ентропію і навчання на основі перцептрона. Моделі, підготовлені за рукописним текстових корпусів вручну, доступні на різних мовах.

#### **1.1.2.1 Intelligent Amplification for Cybercrime (IAC)**

Проект Intelligence Amplification for Cybercrime (IAC) [10] - це проект, виконаний в Університеті Утрехта (Нідерланди), метою якого є поліпшення онлайн-прийому кримінальних скарг і подальші розслідування за тематикою електронної злочинності і кіберзлочинність для національної поліції Нідерландів. Діалоговий модуль, що включає технології NLP, дозволяє проводити міжетнічні діалоги між заявниками та програмними агентами для виявлення злочинців. OpenNLP використовується для розпізнавання іменованого об'єкта в модулі діалогу, зокрема для класифікації таких слів, як особисті імена та імена організацій. Визнані іменовані сутності потім асоціюються з онтологічними поняттями і відносинами. Більш детальна інформація наведена в [11].

#### **1.1.2.2 Vico-Calegari**

У цьому документі [12] представлена архітектура програмного забезпечення для підтримки анонімності документів, тобто для заміни конфіденційних даних таким чином, щоб зберегти конфіденційність документів без зміни їх цінності або корисності. OpenNLP інтегрований в систему під назвою MultiNER адаптер розпізнає іменовані об'єкти для анонімізації, наприклад, осіб або географічних місць, в документах юриспруденції, опублікованих Апеляційними судами та Верховним судом Уругваю.

#### **1.1.2.3 Schilder**

Цей документ [13] являє собою прототип для ідентифікації відповідних подій у Кодексі Сполучених Штатів по імміграційній громадянства США і зв'язування їх з тимчасовою

інформацією, *наприклад*. він вступив в *Сполучені Штати* до 31 грудня 2005 року, завдання, яка вважається вирішальною для автоматичної обробки юридичних документів, які зазвичай містять багато часової інформації (див. [14]). У прототипі openNLP використовується для виконання тегів POS, розщеплення речень і дрібного аналізу на вхідному тексті до того, як ad-hoc система на основі правил аналізованого тексту розпізнає події, часові обмеження і пов'язує перше з останнім.

### 1.1.3 Spacy и TensorFlow

SpaCy - це статистичний інструмент для великомасштабних завдань витяга інформації. Він добре відомий своєю швидкістю при аналізі дуже великого текстового введення. SpaCy розроблено також для реальних і промислових сценаріїв, і його API є простим і продуктивним. Його довгий список опцій включає в себе синтаксичну сегментацію речень, таргетування на частини мови, розпізнавання імен і найпростіших образів, легку інтеграцію глибокого навчання, статистичні моделі для англійської і німецької мов та найсучаснішу швидкість. В деталях Spacy має точність 92,8, а рівень визначення - 94,44 .

SpaCy також легко інтегрується з TensorFlow, бібліотекою програмного забезпечення з відкритим вихідним кодом для чисельного розрахунку з використанням графіків потоку даних. У TensorFlow вузли в графіку являють собою математичні операції, а ребра графа являють собою масиви багатовимірних даних (тензорів), що передаються між ними.

#### 1.1.3.1 Регулювання і відтворення EuroVoc з правовими таксономіями

UNITO впровадив систему Запитання-Відповідь (Question Answering, QA), основу на мережі пам'яті, яка перевіряє розуміння машиною юридичного тексту і визначає, чи є відповідь на питання правильним або неправильним, з урахуванням деяких вихідних знань. UNITO також підготував для цього завдання справжній іспит на право юриспруденцією [15].

QA слідує за процесом навчання людини, тобто удосконалює пам'ять і узагальнює нові події. Автори [16], використовуючи глибокі нейронні мережі, домоглися 100%-ної точності деяких завдань. Тим не менш, використовувався синтетичний набір даних, і оцінки перевіряли здатність моделей надавати фактоідні відповіді на питання про те, де, коли і хто про об'єкт. Впроваджена система відповідає на наступне питання: «чи Можемо ми використовувати глибокі методи навчання для отримання трансферного навчання при проходженні питання-відповідь (PQA) визначаючи схожі шаблони?». При передачі навчання ми маємо на увазі

процедуру узагальнення, згідно з якою запропонована модель може передавати приховані факти з сценарію в аналогічні сценарії.

Deep Networks може самостійно вивчати семантичне подання з тексту. Рекурентні нейронні мережі (RNN) [17] мають спокуси, які мають петлі, додаючи зворотний зв'язок і пам'ять до мереж з плином часу. Тим не менш, пам'ять RNN невелика, а також недостатньо ізольована для зберігання інформації про далекій відстані. [18] запропонували MemN в якості рішення, яке використовується в цій системі. MemN складаються з 4 одиниць, тобто блоків вводу I, блоку узагальнення G, блоку виводу O і блоку відповіді R, який генерує подання Output у будь-якому заданому форматі. Тривала короткострокова пам'ять (LSTM) [19] - це особливий тип RNN, який стійкий до проблеми зникнення градієнта.

### 1.1.3.2 Schrading

Робота [20] спрямована на аналіз домашнього насильства з використанням SpaCy відслідковує дані соціальних мереж. Соціальне середовище і соціальні мережі відіграють важливу роль в нашій країні. Публічно доступні повідомлення на таких сайтах, як Twitter, Reddit, Tumblr і Facebook, можуть містити глибоко особисті дані про життя користувачів. Проблеми зі здоров'ям, сімейні проблеми, оповідання про знущання і будь-яку кількість інших проблем, з якими стикаються люди кожен день, докладно описані в Інтернеті. NLP і машинне навчання можуть допомогти процесу аналізу зрозуміти суспільні проблеми і проблеми суспільної охорони здоров'я. Це дозволяє збирати і аналізувати дані, які можуть пролити світло на соціально-важливі проблеми .

В роботі [20] вивчалася ефективність класифікаторів, які виявляють текст, який обговорює зловживання, щоб підкреслити динаміку образливих стосунків. Аналіз виявив мікроповіді обґрунтовують причини для втручання, щоб зупинити образливі відносини. Цей тип методу може бути також прийнятий у юридичному контексті.

### 1.1.4 Gensim

Gensim - це масштабована, надійна і незалежна платформа python для реалізації неконтрольованого семантичного моделювання з простого тексту, такого як імовірнісний латентний семантичний аналіз (pLSA) і Latent Діріхле Allocation (LDA). Gensim можна інтегрувати з іншими бібліотеками python, такими як NLTK (Natural Language Toolkit) для виконання завдань попередньої обробки NLP. Gensim забезпечує реалізацію tf-idf (загальний

ваговий коефіцієнт при обробці тексту), word2vec (дрібні моделі нейронних мереж, які викликають вбудовування слів) та тематичних алгоритмів моделювання.

#### **1.1.4.1 Автоматичне виявлення трансферів директив Європейського союзу**

У рамках проекту MIREL впроваджено систему, засновану на методах подібності семантичного тексту, щоб автоматично визначати перенесення директив Європейського Союзу (ЄС) у національний закон. В даний час Європейська комісія (ЄК) вдається до трудомістким і дорогим ручним методам, таким як дослідження перевірки відповідності та юридичний аналіз для визначення національних заходів транспозиції. Використовуючи як лексичні, так і семантичні методи подібності і доповнюючи їх знаннями від EuroVoc для визначення транспозицій. Такі системи можуть використовуватися для ідентифікації перенесених положень як ЄС, так і звичайних юристів.

UNITO розробив системи семантичного подібності для автоматичного визначення перенесення директив ЄС у національне законодавство при співпраці з APIS. UNITO розробив систему семантичного подібності, в той час як APIS надає такі ресурси, як директивні документи у форматі XML, і корисний тезаурус і словники, які інтегровані в систему. APIS також розробляє платформу для оцінки та візуалізації результатів, створених Директивами системи транспонування. Поточна система використовує прихований семантичний аналіз та знання з тезаурусу EuroVoc для визначення транспозицій. Отримані транспозиції порівнюються з золотим стандартом або оцінюються експертами APIS для розрахунку точності системи. Ця система буде служити юридичним інструментом пошуку інформації для підтримки юристів і фахівців ЄС, які здійснюють транскордонні юридичні дослідження.

Ефективна трансформація директив Європейського союзу (ЄС) на національному рівні має важливе значення для досягнення цілей договорів і безперебійного функціонування ЄС. Держави-члени несуть відповідальність за правильне і своєчасне виконання директив. Європейська комісія (ЄК) відповідає за моніторинг національних реалізацій, щоб забезпечити їх дотримання законодавства ЄС. Заходи по транспозиції, прийняті державами-членами в національному законодавстві для досягнення цілей цієї директиви, відомі як національні заходи по здійсненню (national implementing measures, NIM). Комісія контролює NIM (повідомлені державами-членами), щоб забезпечити, щоб держави-члени прийняли належних заходів для досягнення цілей цієї директиви. Заходи, прийняті Комісією для моніторингу NIM, включають в себе таблиці перевірки відповідності та кореляції [21]. Комісія аутсорсирує моніторинг NIM субпідрядникам та юридичним консультаційним фірмам. Дослідження перевірки відповідності, проведені групою компетентних експертів з правових питань, включають в себе юридичний

аналіз та таблиці відповідності. У таблицях узгодження вказані конкретні положення NIM, які реалізують конкретну статтю директиви. Держави-члени готують таблиці кореляції, щоб гарантувати, що директива повністю перенесена. Вони визначають конкретні положення NIM для кожної статті директиви в табличному форматі. Таблиці кореляції, як правило, недоступні для громадськості, оскільки вони направляються державами-членами Комісії в рамках конфіденційної двосторонньої біржі. Не існує узгодженого формату або обов'язкового змісту для кореляційних таблиць.

Система, введена UNITO, являє собою першу роботу щодо автоматичного виявлення транспозиції директив ЄС. Метою визначення конкретних положень NIM, які переносять конкретну статтю. UNITO порівнює результати як з лексичними, так і з семантичними методами подібності з їх відповідними NIM, оцінюючи їх за допомогою золотого стандарту (таблиці кореляції) з хорошою продуктивністю.

В деталях, система використовує косинус-векторну модель подібності (метод лексичної подібності), щоб виявити транспонування положень з схожими словами. Латентний семантичний аналіз (метод семантичного подібності), який використовується для виявлення транспонуючих комісій з однаковою семантикою, але з різними формулюваннями. Попередня обробка включала видалення пунктуації, перетворення в нижній регістр і токенизацію. Подальші стоп-слова були видалені, як і тегери частин мови (POS-теггер), який використовувався для фільтрації іменників, дієслів і прикметників з решти набору токенів. Знаки, отримані після попередньої обробки, були збагачені знаннями, отриманими від EuroVoc, багатомовного тезауруса Європейського Союзу. Токени в корпусі були збагачені синонімічним і майже синонімічним терміном згідно з відношенням еквівалентності EuroVoc [22]. Згодом набір нових жетонів зводиться до зменшення флексивних форм слів. Кожне положення корпусу представлено в bag-of-words форматі. Це список кожного сертифіката та його рахунок в конкретній комісії. Крім того, система використовує схему зважування по частотній частоті (tf-idf) для всієї комісії. Він реалізований у латентному семантичному аналізі (LSA), застосовуючи сингулярною декомпозицію значення (SVD) до матриці маркерів надання TF-IDF. SVD розкладає матрицю tf-idf на окремі матриці, які фіксують схожість між токенами і комісіями в різних вимірах у просторі [23].

### **1.1.5 Boxer/CGG parser**

Boxer/CGG-парсер - синтаксично-семантичний інструмент NLP, який включає комбінаторний алгоритм категоріальною граматики (CGG) та синтаксичний семантичний інтерфейс від структур CGG до семантичних уявлень у структурах подання дискурсу (DRS) .

Грамматика CGG є складовою граматикою, яка описує синтаксичні типізовані категорії лексичних предметів разом з їх комбінацією. Наприклад, (багато) дієслова пов'язані з синтаксичною категорією NP/S, вказуючи, що дієслово може поєднуватися з іменниками (NP), щоб побудувати речення (S). У Voxer синтаксичний аналізатор пов'язаний з 1:1 з формальним семантичним перекладом, вираженим в лямбда-численні, так що семантичні диференціювання слідує за структурою синтаксичного дерева. Вони переводяться в DRT, семантичні представлення яких в основному є виразами FOL для подання дискурсів, включаючи займенникові анафори і дискурсні відносини. Синтаксичні і семантичні диференціювання можуть бути задані для довгих і складних речень, а також для дискурсів, які є послідовностями речення.

#### **1.1.5.1 Витяг деонтичних правил з технічних документів**

Незважаючи на численні підходи до вирішення проблеми переходу від природної мови до правової ситуації, результати все ще незадовільні, і це залишається серйозною відкритою завданням. Партнер INRIA і партнер Data61 у співпраці з FBK-Trento (Італія) запропонували підхід до об'єднання методів обробки природної мови з витягом правил з юридичних документів. Точніше, разом з синтаксичним викладом правил з юридичних текстів і логічним витягом залежностей між шматками таких текстів. Такий комбінований підхід призводить до потужного рішення для видобування машиночитаних правил з юридичних документів.

У цій роботі бібліотека Stanford NLP використовується для аналізу природної мови документа для вилучення їх граматичного подання, а для аналізу логічних залежностей між фрагментами тексту документа використовується інструмент Combinatory Categorical Grammar (CCG), включаючи фреймворк Voxer. Більш детальна інформація про цьому підході міститься в [24].

#### **1.1.5.2 Wyner**

У роботі [25] Voxer/CGG використовувався для аналізу British Nationality Act 1981 (ЧАСТИНА I), з тим щоб визначити склад юридичних текстів. Корпус використовувався для ідентифікації кількох мовних явищ в юридичних текстах, щоб пропагувати подальші дослідження NLP, що застосовуються в правовій сфері. Прикладами таких лінгвістичних явищ є узагальнені квантори, інтенціональність і спільність. Крім того, було доведено, що багато випадки складних юридичних питань в даний час не можливо вирішити з допомогою Voxer, так що для вирішення цієї проблеми необхідно додаткове посилення цього інструменту.

### **1.1.6 JFLEX**

JFlex - це генератор лексичного аналізатора (також відомий як генератор сканера) для Java, написаний на Java. В якості введення вводиться специфікація з набором регулярних виразів і відповідних дій. Він генерує програму (лексер), яка читає дані вводу, зіставляється з регулярними виразами в src-файл і запускає відповідну дію, якщо регулярний вираз відповідає. Лексери JFlex базуються на детермінованих кінцевих автоматах (DFA).

#### **1.1.6.1 VO-ECLI**

JFLEX використовується для ідентифікації національних і європейських посилань в юридичних документах VO-ECLI. Проект ECLI (VO-ECLI) - це проект з участю шести держав-членів (Італія, Греція, Хорватія, Естонія, Бельгія, Нідерланди, Німеччина, Чехія, Іспанія, Румунія) та з метою подальшого поліпшення доступності прецедентного права. У проєкті UNITO і UNIBO підтримують створення основи для ідентифікації та побудови ідентифікаторів ECLI.

### **1.1.7 Parse-IT**

Parse-IT є складовою частиною платформи Digital Legislation Platform. Parse-IT - це пропріетарная веб-система, яка може бути прочитана і зрозуміла як захисна деонтична логіка [26]. Інструмент приймає нормативний текст і ідентифікує логічну структуру норм і представляє їх у вигляді набору правил. Таким чином, Parse IT включає спеціалізований редактор правил для поліпшення, виправлення і тонкої настройки правил, витягнутих з юридичних документів. Потім правила можуть бути використані, щоб знаходити різні вирішення юридичних завдань, таких як дотримання бізнес-процесів [27] і аналіз контрактів [28].

#### **1.1.7.1 PermitME**

Parse-IT був використаний у проєкті PermitME у співпраці з Австралійським податковим управлінням, спрямованим на створення цифрового помічника лабіринту законодавства для керівництва SME, який застосовується до їхнього бізнесу, визначення які дозволи і ліцензії

необхідні для бізнесу, і які вимоги до подання заявок на такі дозволи і ліцензії повинні задовольнятися .

### **1.1.7.2 Regorous**

Parse IT був інтегрований в Data61 набір відповідності бізнес-процесів Regorous. Платформа цифрового законодавства призначена для використання в якості основи для оцифровки і формалізації законодавства і створення API для витягів та обґрунтування з формалізованого законодавства для створення правових додатків .

### **1.1.8 TULE parser**

Парсер TULE (Туринський університет мовного середовища) [4] був розроблений на факультеті комп'ютерних наук Університету Турина, і він широко використовується партнерами UNITO і UL в MIREL для аналізу юридичного тексту. Причина в тому, що він відрізняється від більшості інших існуючих парсерів, наприклад тим, які перераховані вище, TULE засновані на правилах, тому можна виправити аналіз NLP без навчання парсера в нових корпусах. TULE включає в себе Tokenizer, роздільник пропозицій, POS Tagger і аналізатор залежностей. TULE працює на італійській, англійській та інших мовах, які не мають відношення до MIREL. Для італійської мови він був розміщений на корпусі TUT, який є найбільшим синтаксичним корпусом (treebank), він містить понад 3500 пропозицій, анотованих за форматом залежності TULE. Заяви TUT були взяті з газети, італійського цивільного кодексу, записів у Вікіпедії. Для французької та англійської мов О'тул був прочитаний в JRC-Acquis, Загальної декларації прав людини (UDHR), Parallel Corpus Європейського парламенту [29] та інших .

#### **1.1.8.1 ProLeMAS**

Проект ProLeMAS (мова програмування в нормативних мультиагентних системах) - це індивідуальний проект, виконаний Лівіо Робальдо в університеті Люксембурга. Проект підтримується індивідуальною стипендією Марії Скловської-Кюрі. Однією з цілей проекту є впровадження набору інструментів NLP для реалізації розпізнавання іменних об'єктів за допомогою аналізу залежностей, а також для визнання зобов'язань, прав і т. д. в директивах ЄС. TULE - аналізатор залежностей, що використовується в проекті .



### **1.1.8.2 DAPRECO**

Проект DAPRECO (DAta PROtection REgulation COmpliance) є проектом CORE, який був відкрит для фінансування у листопаді 2016 року. Проект розпочався в лютому 2017 року і триватиме два з половиною роки. Університет Болоньї є зовнішнім партнером проекту. Інструменти NLP, розроблені в ProLeMAS, а також парсер TULE, планується прив'язати до анотатора LIME, розробленого в Університеті Болоньї, для створення напівавтоматичного підходу до анотації юридичних документів. См. [30].

### **1.1.8.3 Eunomos**

Eunomos - це передова система управління правовими документами, що ґрунтується на законодавчих XML-уявленнях законів, які автоматично витягуються з інституційних законодавчих порталів. Система описана в [31]. Парсер TULE використовується в Eunomos, щоб ідентифікувати юридичні документи, що зберігаються в системі. Комерційна версія Eunomos під назвою Menslegis запатентована і поширена компанією Nomotika SRL .

### **1.1.8.4 OpenSentenze**

Проект OpenSentenze спрямований на публікацію прецедентного права з боку італійських судів як відкритих даних. Для цього необхідно, щоб прецедентне право було анонімним: імена сторін, а також будь-яка інша інформація для ідентифікації сторін (адреси, дати народження тощо) повинні бути приховані. Зверніть увагу, що, з іншого боку, у документі можуть з'являтися власні імена суддів і юристів. Парсер TULE використовується напівавтоматичним способом разом з анотатором LIME, розробленим в Університеті Болоньї. Парсер TULE генерує XML-файли в Akoma Ntoso, де пропонуються анонімні об'єкти. Людський анотатор перевіряє слова або додає відсутні анотації.

### **1.1.9 SPeLT**

CIRSFID (UNIBO) розробив спеціальний (семантичний аналізатор правового тексту), інструмент для парсингу і аналізу юридичних текстів. В даний час написано два основних інструменту для синтаксичного аналізу; SPeLT-ref і SPeLT-struct. Мета SPeLT-ref полягає у визначенні юридичних посилань у судових рішеннях та інших юридичних документах, в той час як мета SPeLT-struct полягає у визначенні логічної структури правових документів. Таким

чином, SPELLED надає набір правил, званих SPELT-anony, які можуть використовуватися стороннім програмним забезпеченням, щоб анонімізувати певні сутності в судженнях і інших юридичних документах. Нарешті, SPELT поставляє інструмент розмітки, промаркований міткою, який може використовувати результати аналізатора для розмітки неструктурованого документа за допомогою AkomaNtoso. Для досягнення цієї мети SPELT-mark може використовувати результати, повернуті SPELT-struct і SPELT-ref, але також може використовуватися стороннім програмним забезпеченням.

### **1.1.9.1 EuCases**

Проект EuCases був проектом FP7 за участю партнерів MIREL APIS, UNITO, Nomotika і UNIBO. Spelt NLP використовуються в редакторі LIME для маркування італійських актів, щоб продемонструвати ефективність Akoma Ntoso для подання директив ЄС, що опубліковані в «Official Gazette». Інструмент орфографії знаходить в простому тексті основну правову інформацію в правових актах ЄС таких, як: структура документа, номер документа, тип документа, щодо повноваження, які випускають в дію, нормативні посилання, дати, особи, організації, місця, ролі.

### **1.1.9.2 Проект Швейцарської канцелярії**

Проект був частиною угоди між UNIBO і Швейцарської канцелярією. Spelt NLP були використані всередині редактора LIME для розмітки документів публікуються Федеральна канцелярія Швейцарії таких як, акти, референдуми, зведений код, щоб продемонструвати ефективність Akoma Ntoso для управління робочим процесом публікації в багатомовному контексті (французька, німецька, італійська). Інструмент базується на використанні таблиць стилів Word, які використовуються кінцевими користувачами, він може виявляти основну юридичну інформацію в швейцарських документах як: структура документа, номер документа, тип документа, щодо повноваження, які випускають дію, нормативні посилання, дати, осіб, організації, місця, ролі .

### **1.1.9.3 Проект верховного касаційного суду**

Проект був частиною угоди між UNIBO Касаційним судом Італії. Інструменти Spelt NLP використовувалися в редакторі LIME для маркування італійських актів, щоб продемонструвати ефективність Akoma Ntoso для публікації італійських юридичних документів, опублікованих у

Official Gazette. Інструмент Spels виявляє звичайного тексту основну інформацію про юридичну інформації в італійських правових актах таких, як: структура документа, номер документа, тип документа, орган, який видає акт, нормативні посилання, дати, особи, організації, місця, ролі.

#### 1.1.9.4 Проект FAO

Проект є частиною угоди між UNIBO та Продовольчою і сільськогосподарською організацією Об'єднаних Націй(FAO). Інструменти Spelt NLP використовуються в редакторі LIME для маркування документів FAO(стандарти, основні тексти, дозвіл), щоб продемонструвати ефективність Akoma Ntoso для моделювання робочого процесу створення документів та їх публікації. Інструментарій витягує з простого тексту, що містить основну документальну інформацію правових актів, таку інформацію: структуру документа, номер документа, документ про повноваження, дату, особи, організації, місцезнаходження, ролі, ключові слова .

#### 1.2 Результати

В результаті роботи було з'ясовано які основні NLP інструменти, використовуються співтовариством правової інформатики для вилучення необхідної інформації з юридичних текстів з допомогою різних методів, ці інструменти наведені в табл. 1.1.

Таблиця 1.1 - Інструменти аналізу юридичних документів

<b>Найменування інструменту</b>	<b>Framework</b>	<b>Інструменти NLP</b>
Автоматична анотація семантичних об'єктів і ролей	Stanford parser	Синтаксичні аналізатори, POS
Пошук і класифікація іменованих об'єктів в юридичних документах	Stanford parser	Синтаксичні аналізатори, NER
Автоматична анотація семантичних об'єктів і ролей	Stanford parser	Синтаксичні аналізатори, POS
Пошук і класифікація іменованих об'єктів в		Синтаксичні аналізатори, NER

юридичних документах		
Huang		Синтаксичні аналізатори
Wyner		Синтаксичні аналізатори
Intelligent Amplification for Cybercrime (IAC)	Apache OpenNLP	NER
Vico-Calegari		NER
Schilder		POS
Регулювання і відтворення EuroVoc з правовими таксономіями	Spacy и TensorFlow	Питання-Відповідь, Семантична анотація
Schrading		Синтаксичні аналізатори, Семантична анотація
Gensim		Семантична анотація
Автоматичне виявлення трансферів директив Європейського союзу		Синтаксичні аналізатори, Семантична анотація
Витяг деонтичних правил з технічних документів	Boxer/CGG parser	Синтаксичні аналізатори, Семантична анотація
Wyner		Синтаксичні аналізатори, Семантична анотація
BO-ECLI	JFLEX	NER
PermitME	Parse-IT	Синтаксичні аналізатори, Тематичне моделювання
Regorous		Синтаксичні аналізатори, Тематичне моделювання
ProLeMAS	TULE parser	Синтаксичні аналізатори,NER, Тематичне моделювання, Семантична анотація
DAPRECO		Синтаксичні аналізатори, Семантична анотація
Eunomos		Синтаксичні аналізатори,Семантична анотація
OpenSentenze		Синтаксичні аналізатори,Семантична анотація
EuCases	SPeLT	Синтаксичні аналізатори,Семантична анотація,NER
Проект Швейцарської канцелярії		Синтаксичні аналізатори,Семантична анотація,NER
Проект верховного касаційного суду		Синтаксичні аналізатори,Семантична анотація,NER
Проект FAO		Синтаксичні аналізатори,Семантична анотація,NER

Як видно з розділу 1.1 та табл. 1.1, головним чином використовується інструмент Stanford Parser, статистичний інструмент, який виконує розбір текстів, написаних природною мовою.

Інші інструменти статистичного NLP є Apache OpenNLP, Spacy, TensorFlow, і Gensim .

Крім статистичних інструментів, співтовариство правової інформатики, також застосовує інструменти засновані на правилах .

### 1.3 Завдання проекту

Наведено та стисло описані основні інструменти NLP, більшість з яких дійсно є інструментами багатоцільовими, тобто конкретно не використовується для обробки тільки юридичних текстів, а також конкретні проекти та науково-дослідної діяльності правової інформатики, в яких вони були використані .

Галузь права завжди була привабливою для синтаксичної та семантичної технології з-за її важливості для суспільства у процесах глобалізації та спільних ринків, а також рішення задач формалізації і використання конкретних мов. З цієї причини нещодавно була організована серія семінарів, присвячених правової інформатики та суміжних тем; прикладами є ICAIL (International Conference on Artificial Intelligence and Law, Міжнародна конференція по штучному інтелекту та праву), JURIX(міжнародний семінар з юриспруденції) і JURISIN(міжнародний семінар з юриспруденції). Крім того, нещодавно були профінансовані кілька дослідних проектів в юридичній області ЄС та аналогічних інститутів, серед яких «MIREL: Mining and Reasoning with Legal texts(Збирання та осмислення юридичних текстів)».

Розробка технологій NLP(Natural Language Processing, Обробка природної мови), семантичних технологій для автоматичного аналізу та індексації великих даних, вільно доступних в Інтернеті, створила можливості для створення нових підходів підвищення ефективності, зрозумілості і послідовності правових систем. Семантичний аналіз спрямовано на зв'язування синтаксичних елементів - які можуть бути фразами, реченнями, реченнями, параграфами і цілісними документами, до їх значень в певній галузі, включаючи значення, характерні для правової інформації. З одного боку, в останні роки Європейський Союз надав величезну кількість ресурсів на законодавство ЄС на багатьох мовах (таких як EuroParl, JRC тощо), з іншого боку, зріла технологія NLP і Semantic Web забезпечує якісні інструменти:

для формалізації юридичних даних у формі інформаційних онтологій;

для автоматизації процесу витяг значних даних з юридичних документів;

для представлення юридичних документів у вигляді зв'язаних даних у форматі RDF.

Це буде сприяти вирішенню юридичних завдань, таких як поліпшення можливостей пошуку, перевірка відповідності та підтримка прийняття рішень, а також краще уявлення правової інформації професійним і непрофесійним зацікавленим користувачам.

В Україні одною з основних установ що займаються проблемами правової інформатики є Науково-дослідний інститут інформатики і права Національної академії правових наук України

створений в 2001 році. Відповідно до Статуту головними завданнями Інституту є: проведення комплексних фундаментальних і прикладних досліджень з метою одержання нових знань з теоретико-правових проблем розбудови інформаційного суспільства та інтеграції України у світовий інформаційний простір, в галузі інформаційного права, правової інформатики та інформаційної безпеки, розробки і впровадження у практику сучасних методів і засобів обробки даних, електронних систем і баз даних в галузі держави і права, сприяння державним органам у розвитку України як демократичної, правової, соціальної держави.

Основні задачі, які вирішуються за допомогою засобів перелічених вище, є: частиноречна розмітка, розпізнавання ліц, розбір, сегментування тексту, в розділі моделювання, екстракція ключової фрази, семантична розмітка і запитання-відповідь.

Засновані на правилах інструменти, як правило, мають кращі характеристики, ніж статистичні в основному для завдань, де це має вирішальне значення для пошуку конкретної інформації.

У правової інформатики, одна з основних завдань, що потребує системи на основі правил, є пошук посилань на правовий документ у інших правових документах, які можуть розглядатися в якості конкретної задачі іменування сутностей.

Аналіз існуючих інструментів доказує важливість та актуальність для світового суспільства систем аналізу юридичних текстів. Це доводиться не тільки різноманітністю наукових робіт та конференцій с питань правової інформатики, але також і фінансовою підтримкою науковців, працюючих в цієї галузі, яка виражається в грандах на розробку та впровадження інструментів аналізу.

На мій погляд, така увага суспільства до цієї наукової тематики, пов'язана з великими очікуваннями від інструментів правової інформатики, які можуть допомогти створювати прозоре законодавство, уникати матеріальних втрат у зв'язку з правовою неосвіченістю, протидіяти корупції в органах судової та державної влади.

Якщо уважно подивитися на системи правової інформатики, описані в першому розділі, то можна помітити, що в основному розробники отримують юридичну інформацію з юридичних документів з метою вдосконалення систем пошуку, з метою зручності вилучення та зберігання юридичної інформації в інформаційних базах, тобто створюються інструменти, що допомагають в роботі кваліфікованих юристів.

Хоча ці системи допомагають орієнтуватися в законодавстві і отримувати інформацію, їх загальна корисність обмежена через те, що основна увага приділяється термінологічним питанням, не звертаючи уваги на семантичні аспекти, які допускають юридичні міркування [32]. Але саме семантичні аспекти, що допускають юридичні міркування, і створюють правила,

які використовуються для керівництва проведенням соціальної взаємодії. Ці правила вказують, які дії заборонені, які можливі і які обов'язкові.

Очевидне завдання як раз і полягає в отриманні таких правил з текстів юридичних документів. Для того, щоб зробити це, ми повинні усунути розрив між джерелами природної мови (NL) і машиночитаним поданням правил.

Усунення такого розриву підвищить ефективність аналізу законодавчих текстів законодавства на слов'янських мовах, дозволить створити інструменти, які можуть допомогти створювати прозоре законодавство, допоможуть в пошуку протиріч у законодавчих нормах, дозволять уникати матеріальних втрат у зв'язку з правовою неосвіченістю та протидіяти корупції в органах судової та державної влади.

Таким чином розробка є актуальною.

**Задля вирішення задачі підвищення ефективності аналізу законодавчих текстів законодавства на слов'янських мовах в цій роботі повинні бути виконані наступні завдання:**

1. Вибрати й описати технологію для створення логічного аналізатора юридичних документів.
2. Провести синтаксичний аналіз законодавчого тексту.
  - 2.1. Визначити необхідні програмні засоби для проведення синтаксичного аналізу.
  - 2.2. З допомогою програмних засобів провести синтаксичний аналіз декількох статей Конституції України.
  - 2.3. Зберегти результати аналізу в інформаційній базі.
  - 2.4. Відобразити результати аналізу у вигляді діаграми окремо для кожної статті Конституції.
3. Провести семантичний аналіз законодавчого тексту з використанням результатів що отримані на першому етапі.
  - 3.1. Написати алгоритм, який дозволяє визначати суб'єкти правовідносин.
  - 3.2. Написати алгоритм, який дозволяє визначати об'єкти правовідносин.
  - 3.3. Вказати суб'єкти та об'єкти правовідносин на діаграмі створений на попередньому етапі.

## РОЗДІЛ 2

### ВИБІР ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ЛОГІЧНОГО АНАЛІЗАТОРА ЮРИДИЧНИХ ДОКУМЕНТІВ

Якщо уважно подивитися на системи правової інформатики, описані в першому розділі, то можна помітити, що в основному розробники отримують юридичну інформацію з юридичних документів з метою вдосконалення систем пошуку, з метою зручності вилучення та зберігання юридичної інформації в інформаційних базах, тобто створюються інструменти, що допомагають в роботі кваліфікованих юристів.

Хоча ці системи допомагають орієнтуватися в законодавстві і отримувати інформацію, їх загальна корисність обмежена через те, що основна увага приділяється термінологічним питанням, не звертаючи уваги на семантичні аспекти, які допускають юридичні міркування [32]. Але саме семантичні аспекти, що допускають юридичні міркування, і створюють правила, які використовуються для керівництва проведенням соціальної взаємодії. Ці правила вказують, які дії заборонені, які можливі і які обов'язкові.

Очевидне завдання (що було з'ясовано в першому розділі цієї роботи) полягає в отриманні правил, які використовуються для керівництва проведенням соціальної взаємодії з текстів юридичних документів. Для того, щоб зробити це, ми повинні усунути розрив між джерелами природної мови (NL) і машиночитаним поданням правил.

Величезний поштовх для розвитку систем аналізу природної мови дає еволюція штучних нейронних мереж, які широко використовуються для аналізу та структурування NL. Зокрема за останній рік штучні нейронні мережі показали чудові результати при проведенні глибокого синтаксичного аналізу NL.

Інструменти створені на підставі таких нейронних мереж дозволяють отримувати структуровані дані, які містять лемми або основи словникової форми, списки морфологічних ознак, а також будує відносини залежностей між різними словами і частинами мови. Таким чином якісний синтаксичний аналіз стає відправною точкою в задачах розуміння сенсу, закладеного в текстах юридичних документів.

Наступною дією, спрямованою на розуміння сенсу юридичних документів, повинен бути семантичний аналіз.

У сучасному розумінні семантичний аналіз полягає у виділенні семантичних відносин, формуванні семантичного представлення текстів. В загальному випадку результатом такого аналізу є отримання семантичної мережі, що має вигляд орієнтованого графа, вершини якого відповідають об'єктам предметної області, а дуги (ребра) задають відносини між ними. Саме погляд на семантичну мережу з боку предметної області і створює різноманітність алгоритмів



семантичного аналізу, різних підходів і варіантів побудови інформаційних баз, що містять дані таких мереж.

Для проектування семантичного аналізатора необхідно чітке розуміння для чого будуть використовуватися результати такого аналізу.

Основним напрямком аналізу юридичних текстів, на мій погляд, має бути створення наборів логічних правил. При цьому правила залежать від складових їх елементів:

- суб'єкти правовідносин;
- об'єкти правовідносин;
- суб'єктивні права;
- юридичні обов'язки і дозволу;
- пільги;
- умови при яких правила поведінки діють;
- заходи заохочення;
- санкції;
- способи дозволу конфліктів;
- механізми підтримки санкцій і т. д.

Семантичний аналіз повинен бути другим етапом перед логічним аналізом юридичних текстів.

І завершальним етапом повинен бути логічний аналіз, який створює набір машиночитаних логічних правил.

Розглядаючи приклад, використаний для демонстрації семантичного аналізатора (*"Кожна людина має право на вільний розвиток своєї особистості, якщо при цьому не порушуються права і свободи інших людей, та має обов'язки перед суспільством, в якому забезпечується вільний і всебічний розвиток її особистості."*), видно що склад правовідносин має умови існування, а саме *"якщо при цьому не порушуються права і свободи інших людей"* - для права, *"і в якому забезпечується вільний і всебічний розвиток її особистості"*, для обов'язків.

Звідси виходить, що для розуміння ЗМІСТУ законодавства, не досить класифікувати і групувати елементи правовідносин, що по суті і робить семантична мережа, необхідно логічно проаналізувати текст і на підставі їх створити логічні машинописні правила, які в подальшому зроблять можливим реалізацію алгоритмів для роботи з такими правилами (система питання-відповідь, система оповіщення настання юридичних подій, система пошуку протиріч між створюваним нормативним актом і існуючим законодавством, тощо)

На мій погляд без створення логічного аналізатора юридичних текстів, ПРАКТИЧНА правова інформатика буде завжди мати обмежені можливості, а значить і обмежену область застосування, що зараз і відбувається.

Для подальшої роботи у цьому напрямку, необхідно вибрати технологію для створення програмного логічного аналізатора юридичних текстів, вихідними даними для якого повинна послужити семантична мережа, а результатом роботи, якого повинні стати набори машинописних правил.

У 2016 році на 22-й Європейській конференції по штучному інтелекту (ECAI 2016) в Гаазі, Livio Robaldo зачитав доповідь "Реалізована логіка введення/виводу - документ з позицією"[32] на мій погляд - це важливий документ відкриває нові можливості при смислового аналізу текстів юридичних документів.

На базі ідей, цього доповіді і пропонується створювати програмний логічний аналізатор.

Сучасні системи правової інформатики використовують інструменти NLP для перетворення, часто напівавтоматичного, юридичних документів в стандарти XML, наприклад такі як Akoma Ntoso, де відповідна інформація помічені [33] [34]. Хоча ці системи допомагають орієнтуватися в законодавстві і отримувати інформацію, їх загальна корисність обмежена через те, що основна увага приділяється термінологічних питань, не звертаючи уваги на семантичні аспекти, які допускають юридичні міркування[32].

Deontic Logic (DL) використовується з 1950-х років як формальний інструмент для моделювання нормативних міркувань у законі [35] [36]. Тим не менш, наступні події в DL направляють абстрактний погляд на закон, як на текст правил з дуже нечіткої зв'язком, які можуть бути вирішені за допомогою рішень, отриманих з літератури по семантиці природної мови (NLS). Більшість поточних речень DL є пропозиціональними, в той час як NLS включає в себе широкий спектр дрібнозернистих лінгвістичних явищ, які вимагають формалізму логіки першого порядку (FOL)[32].

Є прагнення створити логічну структуру, здатну заповнити прогалину між стандартними (пропозиціональними) конструкціями, що використовуються в DL, і багатством NLS. З усіх логічних інструментів (незалежно) запропонованих у літературі по NLS і DL відповідно, вважаємо, що два з них мають фундаментальні переваги: (див. розд. 3.3.1.2) FOL від prof.J.R. Хоббс, призначений для моделювання значення висловлювань NL допомогою упредметнення і (див. розд. 3.3.1.4) логіка Вводу/Виводу (I/O), запропоновану в [37] для моделювання деонтологічних нормативних тверджень[32].

Рейфікація(упредметнення) - це концепція, спочатку введена філософом Д. Девідсоном в [38]. Він дозволяє перейти від стандартних позначень FOL, таких як '(give a b c)', стверджуючи, що 'a' дає 'b' з 'c', інший нотації в FOL '(give' e a b c)', де e - рейфікація для представляємої дії. e' - це термін FOL, що позначає подію яка отримується за допомогою дачі 'a' з 'b' до 'c'. Згідно з [39] e назується «ймовірністю»[32].

З іншого боку, логіка I/O є добре відомим формалізмом у DL[40], завдяки здатності справлятися зі стандартними проблемами в DL, наприклад, суперечливими роздумами [37] і моральними конфліктами [41].

У цій статті представлено можливе злиття логіки I/O і логіки вводу-виводу, яка намагається поєднати їх відповідні переваги. Обмежуючи нашу увагу тільки зобов'язаннями і дозволами, тобто двома основними видами норм [42], залишаючи інші види норм за рамками цієї роботи[32].

Для прикладу обрані директиви ЄС, з яких були обрані зобов'язання (1.a) (Dir. 98/5/ EC) і дозвіл (1.b) (Dir. 2001/110/EC). Імовірно, приклад вирішення досить загальний, щоб охопити представницьку частину законодавства ЄС:

(2.1)

а) *Адвокат, який бажає практикувати в якій-небудь державі-члені, відмінному від того, в якому він отримав свою професійну кваліфікацію, реєструється в компетентному органі в цій державі.*

б) *Коли пекарський мед використовується в якості інгредієнта в складному харчовому продукті, термін «мед» може використовуватися в назві продукту складової їжі замість терміна «пекарський мед».*

## 2.1 Супутні роботи

Деякі підходи до правової інформатики намагаються моделювати в деяких деонтичних умовах пропозиції NL, виходять з існуючих норм, таких як, наприклад, в (2.1). Найбільш представницька робота, можливо, [43]. Інші приклади можна знайти в [44] і [45]. Деякі підходи, наприклад, [46], [47] і [48], серед іншого, формалізують юридичні розуміння через Event Calculus [49], логічний мову, розширює упередження, вводячи спеціальні терміни та предикати для обробки часових точок і періодів часу [50]. Аналогічна обліковий запис була досліджена в [51] в логіці модальної дії[32].

Наскільки нам відомо, підхід, який, мабуть, ближче всього до того, який ми збираємося запропонувати нижче, - це, мабуть, мова Маккарті для юридичного дискурсу (LLD) [52]. LLD сильно спирається на попередні дослідження NLS, він використовує уречевлення і націлений на моделювання існуючого юридичного тексту. [53] показує, як можна отримати структури LLD з федеральних цивільних справ в апеляційних судах США через NLP[32].

Однак LLD дуже нагадує формалізми, зазвичай використовувані в NLS, такі як теорія подання дискурсу (DRT) [54] та семантика мінімальної рекурсії (MRS) [55]. Вони характеризуються тісним взаємозв'язком між синтаксисом та семантикою, у відповідності з

відомим *принципом композиційності* Монтегю, наріжним каменем стандартних формалізмів, що використовуються в NLS[32].

Композиційний принцип призводить до подання, заснованому на вкладеннях подформул всередині логічних операторів, які встановлюють ієрархію серед предикатів. Наприклад, просте речення типу «Джон вважає, що Джек хоче їсти морозиво» може бути представлено за такою формулою[32] (припускаючи де-дикто-інтерпретацію квантора існування):

$$\begin{aligned} & \text{вважати}[\text{Джон}, \\ & \quad \text{хотіти}(\text{Джек}, \\ & \quad \quad \exists x[(\text{морозиво } x) \wedge (\text{їсти } \text{Джек } x)] \\ & \quad ) \end{aligned}$$

В цієї формулі, де *вважати* і *хотіти*, модальні оператори приймають окрему людину в якості першого аргументу, а потім (вбудовані) подформулу в якості другого аргументу. В останній формулі оператор вважає, що ієрархічно спотворює потребу оператора в тому сенсі, що останній відбувається в рамках першого[32].

Однак в [56], [25] [57] було показано, серед іншого, що така архітектура запобігає кільком доступним читанням у NL, і повинні бути запроваджені більш складні оператори, здатні зв'язувати передбачення по ієрархії щоб правильно їх подавати[32].

З цієї причини Хоббс запропонував логіку, де всі формули плоскі, тобто де не існує ієрархії серед предикатів[32].

## 2.2 Логічна схема Хоббса

Проф. J. R. Hobbs визначає широкосмуговий логіки першого порядку (FOL) для NLS зосередження на упредметненні см. [58] та кілька інших ранніх публікацій того ж автора. У Хоббсе можливі *ймовірні* або *фактичні* обставини. Це розходження представлено через предикат *Rehist*, який має місце для можливих реальностей, існуючих у світі. Можливі значення можуть бути вставлені в якості параметрів таких предикатів, як бажання, віра і т.д. Обґрунтування може бути застосоване рекурсивно. Той факт, що «Джон вважає, що Джек хоче їсти морозиво»[32], представлений як:

$$\exists_e \exists_{e_1} \exists_{e_2} \exists_{e_x} [(Rehist\ e) \wedge (\text{вважати}'\ e\ \text{Джон}\ e_1) \wedge (\text{хотіти}'\ e_1\ \text{Джек}\ e_2) \wedge (\text{їсти}'\ e_3\ \text{Джек}\ x) \wedge (\text{морозиво}'\ e_3\ x)]$$

Найважливішою особливістю логіки Хоббса, яка відрізняє її від всіх інших нео-Девідсон підходів, наприклад, LLD, є те, що всі формули є «плоскими» в сенсі, описаному вище. Зокрема, рамкової схемою розрізняються формули, що відносяться до АВох онтології, від тих, які належать до її ТВох. АВох включає тільки кон'юнкції атомних предикатів, які затверджуються на термінах FOL. З іншого боку, ТВох визначає ці предикати в термінах предиката *Rexist* і стандартного FOL. Таким чином моделюються всі логічні оператори, наприклад логічні зв'язки[32]. Наприклад, заперечення моделюється з допомогою предиката *not'*, визначеного в ТВох[32], як:

Для всіх  $e$  і  $e_1$  таких, що виконується  $(not' e e_1)$ , також має місце: (2.2)

$$(Rexist e) \leftrightarrow \neg(Rexist e_1)$$

Якщо  $(not' e e_1)$  істинно, все, що ми знаємо, полягає в тому, що індивіди  $e$  і  $e_1$  пов'язані з допомогою предикації *not'*. Але це нічого не говорить нам про реальне існування  $e$ , або  $e_1$  [32]. Аналогічним чином і маються на увазі «кон'юнктивні» і «імплікативні» відносини, такі, як в (2.3) і (2.4) відповідно виконані (з іншого боку, ми опускаємо диз'юнкцію)[32].

Для всіх  $e, e_1, e_2$  таких, виконується  $(and' e e_1 e_2)$ , також має місце: (2.3)

$$(Rexist e) \leftrightarrow (Rexist e_1) \wedge (Rexist e_2)$$

Для всіх  $e, e_1, e_2$  таких, виконується  $(imply' e e_1 e_2)$ , також має місце: (2.4)

$$(Rexist e) \leftrightarrow ((Rexist e_1) \rightarrow (Rexist e_2))$$

Хоббс і його послідовники реалізують досить великий набір предикатів для обробки складових сутностей, причинності, часу, неможливості, структури подій та інше. Наприклад, [59] пропонує рішення моделювати концесивні відносини, одне з найбільш складних семантичних відносин, що відбуваються в NL, за логікою Хоббса[32].

Значення предикатів обмежена додаванням «схем аксіом». Непрямі обмеження забороняють ілюструвати деталі про всіх предикатах, визначених Хоббс. Можлива схема аксіом для правової області показана в (2.5). (2.5) говорить, що всі юристи - це люди:

Для всіх  $e_1, x$  таких, що мають місце  $(юрист' e_1 x)$ , також має місце: (2.5)

$$\exists e_i \exists e_2 [(imply' e_i e_1 e_2) \wedge (Rexist e_i) \wedge (\text{люди}' e_2 x)]$$

### 2.3 Обробка деонтичних обґрунтованих аргументів у правовій інтерпретації

Основна проблема правової інформатики стосується правильного тлумачення законів у даних ситуаціях, яке залежить від суддів у судах [60]. *Юридична інтерпретація* - добре вивчена тема правової інформатики, напр. [61] та інші. Наприклад, (1.a), в якій мірі ми повинні думати про адвоката, який *хоче* практикувати в державі-члені, відмінному від того, який він отримав у своєї кваліфікації? Під *буквальним* тлумаченням дієслова «хоче», яке може бути прийняте в якості інтерпретації за замовчуванням, адвокат, який просто *говорить* деяким друзям, яких він хотів би зробити, вже порушує норму, якщо він не зареєстрований в компетентному органі. З іншого боку, розумна (прагматична) інтерпретація полягає в тому, що норма порушується тільки в тому випадку, якщо незареєстрований адвокат виконує деякі «формальні» дії, такі як захист кого-небудь у суді. Згідно з нормою, ця дія має бути заблоковано, і адвокат повинен заплатити штраф[32].

До цих пір було запропоновано кілька підходів для обробки багатьох юридичних інтерпретацій в логіці. Недавнім є [62], де пропонується рішення для їх вирішення в «Неможливою деонтической логіці» [44] з допомогою пріоритетних допустимих правил. Пріоритети вводяться для ранжирування доступних інтерпретацій, тобто для вирішення потенційних конфліктів між ними[32].

Слідуючи [62], обробляються множинні правові інтерпретації за допомогою методології Хоббса для вирішення проблеми здійсності, яка, в свою чергу, взята з Circumscriptive Logic [63].

Ідея проста, проілюструємо її прикладом. Той факт, що кожна птиця літає, представляється в FOL як  $\forall x [\text{птиця}(x) \rightarrow \text{літати}(x)]$ . Щоб зробити правило допустимим, ми додамо ще один предикат *normalBF*, в якому говориться, що птахи літають, тільки якщо це «нормально» припустити:  $\forall x [(\text{птиця}(x) \wedge \text{normalBF}(x)) \rightarrow \text{літати}(x)]$ . Додавання того, що ему є нелетаючими птахами, тобто  $\forall x [\text{ему}(x) \rightarrow (\text{птиця}(x) \wedge \neg \text{літати}(x))]$ , не тягне за собою неузгодженість. Це означає, що *normalBF*(*x*) є хибним для кожного ему *x*. У цьому сенсі останнє правило «сильніше», ніж перше. Альтернативно, ми можемо прямо стверджувати, що ему не є «нормальним» відносно властивості літання, тобто  $\forall x [\text{ему}(x) \rightarrow \text{normalBF}(x)]$ . *normalBF* слід вважати істинним, щоб викликати властивість літати на птахів[32].

Аналогічним чином обробляються різні правові інтерпретації «побажань» (1.а). Припустимо за замовчуванням, що якщо адвокат  $x$  говорить, що він буде практикуватися в у державі-члені, то він справді хоче це зробити[32].

Для всіх  $x, y, e_1, e_2, e_3$  що виконується (2.6)  
 (адвокат  $x$ ) $\wedge$ (ДЧ  $y$ ) $\wedge$ (говорити'  $e_1$   $x$   $e_2$ ) $\wedge$ (бажати'  $e_2$   $x$   $e_3$ ) $\wedge$ (практикувати'  $e_3$   $x$ ) $\wedge$ (in  $e_3$   $y$ ),  
 також має місто:

$$\exists_{ei} [(imply' e_i e_1 e_2) \wedge (Rexist e_i)]$$

Щоб зробити (2.6) здійсненним, ми додаємо предикат  $normalSP$ , який заявляє, що вступ дійсний тільки в тому випадку, якщо воно «нормально», щоб зробити його допустимим[32]:

Для всіх  $x, y, e_1, e_2, e_3$  що виконується (2.7)  
 (адвокат  $x$ ) $\wedge$ (ДЧ  $y$ ) $\wedge$ (говорити'  $e_1$   $x$   $e_2$ ) $\wedge$ (бажати'  $e_2$   $x$   $e_3$ ) $\wedge$ (практикувати'  $e_3$   $x$ ) $\wedge$ (in  $e_3$   $y$ ),  
 також має місто:

$$\exists_{ei} \exists_{e_a} \exists_{e_n} [(imply' e_i e_a e_2) \wedge (Rexist e_i) \wedge (and' e_a e_1 e_n) \wedge (normalSP' e_n e_1)]$$

В (2.7) реальне існування  $e_1$  уже недостатньо для того, вже недостатньо для того, щоб тягло за собою одне із  $e_2$ . Для того, щоб дозволити потяг, необхідно і реальне існування  $e_n$ . Тепер суддя може розумно вирішити, що буде ненормально, припускати, що адвокат, який говорить, що він буде практикувати в державі-члені, тягне за собою те, що він «хоче» (в сенсі описаном в (1.а))[32], тобто:

Для всіх  $x, y, e_1, e_2, e_3$  що виконується (2.8)  
 (адвокат  $x$ ) $\wedge$ (ДЧ  $y$ ) $\wedge$ (говорити'  $e_1$   $x$   $e_2$ ) $\wedge$ (бажати'  $e_2$   $x$   $e_3$ ) $\wedge$ (практикувати'  $e_3$   $x$ ) $\wedge$ (in  $e_3$   $y$ ),  
 також має місто:

$$\exists_{e_n^n} \exists_{e_n} [(not' e_n^n e_n) \wedge (Rexist e_n^n) \wedge (normalSP' e_n e_1)]$$

З (3.8) випливає, що у випадку, якщо адвокат  $x$  просто заявляє, що хоче практикувати в державі-члені  $y$ , ми робимо висновок, що  $e_n$  дійсно не існує. Таким чином, з (2.7) вже неможливо вивести, дійсно  $e_2$  існує чи ні[32].

## 2.4 Логіка Вводу/Виводу (I/O)

Логіка вводу/виводу (I/O) була введена в [37]. Вона виходить з вивчення умовних норм. Логіка вводу/виводу- це сімейство логік, так само як модальна логіка - це сімейство систем K, S4, S5 та інше. Однак, на відміну від модальної логіки, яка зазвичай використовує можливу глобальну семантику, логіка вводу-виводу використовує оперативну семантику: I/O задумана як «дедуктивна машина», як чорний ящик, який виробляє деонтичні оператори в якості висновку, коли ми подаємо його фактичні твердження як вхідні дані[32].

Як пояснюється в [40], операційна семантика вирішує відому дилему Йоргенсена [64], в якій грубо сказано, що правильна справжня логічна нормалізація норм неможлива, оскільки норми не мають значень істинності. На думку Йоргенсена, типові проблеми стандартної деонтичної логіки виникають з його теорії теорії істинності, тобто можливої світової семантики. З іншого боку, оперативна семантика прямо дозволяє справлятися з суперечливими роздумами, моральними конфліктами і таке інше[32].

Крім того, логіка I/O є однією з небагатьох існуючих рамок для нормативних міркувань, у яких також були детально вивчено дозволи, а не тільки зобов'язання. Більшість поточних пропозицій спеціально не присвячені розгляду існуючого законодавства, і тому вони в основному зосереджені тільки на зобов'язаннях[32]. Наприклад, в [48], присвячених обробці відповідності бізнес-процесів (BPC), зобов'язання аналізуються детально, в той час як дозволи здебільшого ігноруються, оскільки перші грають роль в BPC, більш помітну, ніж останні[32]. Запис у [48] була нещодавно розширена для обробки дозволів в [65].

В [37] визначено чотири основні логіки вводу-виводу:  $out_1$ ,  $out_2$ ,  $out_3$  і  $out_4$ . Нехай  $L$  - стандартна пропозиціональна логіка,  $O$  і  $P$  - дві підмножини  $L \times L$ , а  $A$  - підмножина  $L$ , тобто набір формул в стандартній пропозиційну логіку. Кожна пара  $(a, b)$  в  $O$  прочитується як «задана  $a$ ,  $b$  обов'язкова», в той час як кожна пара  $(c, d)$  в  $P$  прочитується як «задана  $c$ ,  $d$  дозволена». Пари  $O$  і  $P$  називаються «генераторами» і являють собою «машину дедукції»: всякий раз, коли на вході вводиться одна з лівих сторін (LHS) пар, відповідна права частина (RHS) задається на виході[32].



(2.9) визначає семантику  $out_1, \dots, out_4$ .  $C_n$  є наслідком оператора пропозиційної логіки; він вводить набір формул  $A$  і повертає безліч, відповідне транзитивному замикання всіх формул, які можуть бути пов'язані з  $A$ . Множина формул є повним, якщо воно або максимальньо узгоджено, або одно  $L$  [32].

$$out_1(O, A) = C_n(O(C_n(A))) \quad (2.9)$$

$$out_2(O, A) = \cap \{C_n(O(V)) : A \subseteq V, V \text{ is complete}\}$$

$$out_3(O, A) = \cap \{C_n(O(B)) : A \subseteq B = C_n(B) \supseteq O(B)\}$$

$$out_4(O, A) = \cap \{C_n(O(V)) : A \subseteq V \supseteq O(V), V \text{ is complete}\}$$

В (2.10) повідомляємо про аксіомах, необхідних для визначення систем I/O, що мають семантику від  $out_1$  до  $out_4$ .  $\vdash$  пов'язане відношення пропозиційної логіки [32].

$$SI: \text{ from } (a, x) \text{ to } (b, x) \text{ whenever } b \vdash a. \quad (2.10)$$

$$OR: \text{ from } (a, x) \text{ and } (b, x) \text{ to } (a \vee b, x).$$

$$WO: \text{ from } (a, x) \text{ to } (a, y) \text{ whenever } x \vdash y.$$

$$AND: \text{ from } (a, x) \text{ and } (a, y) \text{ to } (a, x \wedge y).$$

$$CT: \text{ from } (a, x) \text{ and } (a \wedge x, y) \text{ to } (a, y).$$

Аксіоми в (2.10) обмежують генератори, що належать  $O$  і  $P$ . Наприклад, СТ говорить, що у випадку, якщо два генератора  $(a, x)$  та  $(a \wedge x, y)$  належить  $O$ , то також генератор  $(a, y)$  повинен належати  $O$  [32].

Система деривації, заснована на SI, WO и AND, називається *deriv1*. Додавання OR к *deriv1* дає *deriv2*. Додавання СТ к *deriv1* дає *deriv3*. П'ять правил разом дають *deriv4*. Кожен *deriv<sub>i</sub>* є правильним і повним щодо  $out_i$  (див. [37]).

Приклад того, як аксіоми в (2.10) працюють на практиці, наводиться нижче безпосередньо на нашій логіці об'єктів FOL. Як зазначалося вище, виразність логіки I/O, а також одного з її конкурентів, наприклад Imperative Logic (Імперативна логікою) [66], Prioritized Default Logic (Пріоритетною логікою за замовчуванням) [67] і Defeasible Deontic Logic (Неприпустимо деонтична логіка) [44], серед іншого, обмежена пропозиціональним рівнем. З іншого боку,

логіка Хоббса завдяки своїй формальній простоті дозволяє підвищити виразність систем вводу-виводу до рівня першого порядку з невеликими змінами аксіом (2.10)[32].

## 2.5 Об'єднання логіки Вводу/Виводу та логіки Хоббса

Пропозиціональна логіка не володіє достатньою виразністю для представлення реальних зобов'язань і дозволів, таких як (1.a-b). Також необхідні універсальні квантовані змінні і постійні або функціональні терміни[32].

Наприклад, «адвокат» і «держава-член» (1.a) відносяться до кожного адвоката і кожній державі-члену. З іншого боку, вираження «в якому він отримав свою професійну кваліфікацію» повинно бути представлено як функція  $f_1(x)$ , яка, з урахуванням адвоката  $x$ , повертає держава-член, де він отримав свою професійну кваліфікацію. Аналогічним чином вираз «компетентний орган у цій державі» представляється як функція  $f_2(y)$ , яка, враховуючи держава-член  $y$ , повертає компетентний орган в цій державі. Нарешті, 'термін « мед »' і 'термін «пекарський мед»' (1.b) відповідають двом константам FOL -  $T_h$   $T_{bh}$  відповідно, позначаючи два слова[32].

Наші формули - це кон'юнкції Хоббса атомних передбачень, можливо, пов'язані з змінними FOL. Деякі з цих змінних будуть зустрічатися як у LHS, так і в RHS генератора вводу-виводу, в той час як інші будуть відбуватися або в LHS, або в RHS. Змінні, що відбуваються в обох, будуть універсально квантофіцеровані, а ті, які відбуваються в одному з двох, будуть екзистенційно квантуватись. Крім того, ми вимагатимемо, щоб кожна формула логіки об'єкта стверджувала рівно один  $\text{Rexist}$  предикат в основному випадку. Як пояснюється в розділі 3, семантика логіки Хоббса зосереджена на предикаті  $\text{Rexist}$ [32].

Ми додаємо одну конструкцію до синтаксису генераторів: універсальні квантори для зв'язування змінних, що зустрічаються як у LHS, так і в RHS. Ці квантори діють як «мости» між LHS і RHS, щоб «нести» індивідуумів від входу до виходу. Формально наші генератори мають наступний вигляд: LHS  $(x_1, x_2, \dots, x_n)$  і RHS  $(x_1, x_2, \dots, x_n)$  - кон'юнкції предикатів FOL;  $x_1, x_2, \dots, x_n$  вільні у LHS і RHS, але вони зовні пов'язані універсальними кванторами. LHS і RHS, можливо, включають інші екзистенційно квантовані змінні[32].

$$\forall_{x_1} \forall_{x_2} \dots \forall_{x_n} (\text{LHS}(x_1, x_2, \dots, x_n), \text{RHS}(x_1, x_2, \dots, x_n))$$

Цей архітектурний вибір мотивований емпіричним аналізом зобов'язань/дозволів у нашому корпусі директив ЄС. Норми, встановлені в законодавстві, зазвичай зберігаються для всіх членів певного набору осіб, наприклад, набір всіх юристів. З іншого боку, ми не знайшли в нашому корпусі будь-яких зобов'язань або дозволів у формі «Якщо адвокат існує, то він

зобов'язаний зробити певні дії». Це звучить досить інтуїтивно: заяви в законодавстві зазвичай є універсальними твердженнями, тобто вони не належать до окремих конкретних особам[32].

Зауважимо, що в будь-якому разі, до тих пір, поки формули є кон'юнкціями атомних предикатів, можна легко впоратися із зобов'язаннями/дозволами, видаливши екзистенції допомогою сколемізації. Генератор у вигляді  $\exists_x (\text{LHS}(x), \text{RHS}(x))$  може бути замінений  $(\text{LHS}(i), \text{RHS}(i))$ , де  $i$  - постійна FOL сколемізуюча  $\exists_x$ . З іншого боку, генератор у вигляді  $\forall_x \exists_y (\text{LHS}(x, y), \text{RHS}(x, y))$  може бути замінений на  $\forall_x (\text{LHS}(x, f(x)), \text{RHS}(x, f(x)))$ , де  $f$  - функція FOL, сколемізуюча  $\exists_x$ . Екзистенції, що зустрічаються у логічних формулах об'єкта, також можуть бути сколемізовані. Наприклад, генератор в формі  $\forall_x (\exists_y \text{LHS}(x, y), \text{RHS}(x))$  може бути замінений на  $\forall_x (\text{LHS}(x, f(x)), \text{RHS}(x))$ , де  $f$  - функція FOL, сколемізуюча  $E_x$ [32].

Точно так само слід зауважити, що в кінцевих областях універсальні квантори - це всього лише компактний спосіб звернення до всіх індивідів у всесвіті. Ми отримуємо еквівалентну безліч генераторів, підставляючи універсально квантовані змінні з усіма константами, які посилаються на кожного людини у всесвіті. Наприклад, якщо припустити, що всесвіт включає тільки індивідів  $a, b, c$ , генератор  $\forall_x (\text{LHS}(x), \text{RHS}(x))$  еквівалентний безлічі генераторів  $(\text{LHS}(a), \text{RHS}(a)), (\text{LHS}(b), \text{RHS}(b))$  і  $(\text{LHS}(c), \text{RHS}(c))$ [32].

## 2.6 Узагальнення аксіом логіки вводу/виводу

Ми запропонували вище інтегрувати логіку Хоббса в генератори I/O, просто додавши широкомасштабні універсальні квантори до синтаксису генераторів, щоб створити «міст» для «перенесення» умов FOL, відповідних LHS, на вихід. Також аксіоми в (2.10) повинні бути узагальнені відповідно. У цьому розділі показано узагальнення аксіоми СТ. Узагальнення інших аксіом схоже. СТ узагальнюється так само, як в (2.11)[32].

$$\begin{aligned}
 & \text{from: } \forall_{x_1} \dots \forall_{x_n} ( \tag{2.11} \\
 & \exists_{e_{11}} \exists_{y_{11}} \dots \exists_{e_{1i}} [(\text{Rexist } e_{11}) \wedge (\psi'_{11} e_{11} y_{11} \dots y_{1i} x_1 \dots x_n)], \\
 & \exists_{e_{21}} \exists_{y_{21}} \dots \exists_{e_{2j}} [(\text{Rexist } e_{21}) \wedge (\psi'_{21} e_{21} y_{21} \dots y_{2j} x_1 \dots x_n)] \\
 & \text{and: } \forall_{x_1} \dots \forall_{x_n} (\exists_e \exists_{e_{11}} \exists_{y_{11}} \dots \exists_{y_{1i}} \exists_{e_{21}} \exists_{y_{21}} \dots \exists_{y_{2j}} [(\text{Rexist } e) \wedge (\text{and}' e e_{11} e_{21}) \wedge ( \\
 & \psi'_{11} e_{11} y_{11} \dots y_{1i} x_1 \dots x_n) \wedge (\psi'_{21} e_{21} y_{21} \dots y_{2j} x_1 \dots x_n)], \\
 & \exists_{e_{31}} \exists_{y_{31}} \dots \exists_{e_{3k}} [(\text{Rexist } e_{31}) \wedge (\psi'_{31} e_{31} y_{31} \dots y_{3k} x_1 \dots x_n)]
 \end{aligned}$$

$$\begin{aligned} \text{to: } & \forall x_1 \dots \forall x_n ( \\ & \exists e_{11} \exists y_{11} \dots \exists e_{1i} [(Rexist\ e_{11}) \wedge (\psi'_{11} e_{11} y_{11} \dots y_{1i} x_1 \dots x_n)], \\ & \exists e_{31} \exists y_{31} \dots \exists e_{3k} [(Rexist\ e_{31}) \wedge (\psi'_{31} e_{31} y_{31} \dots y_{3k} x_1 \dots x_n)] \end{aligned}$$

Прикладом може служити: «Кожен адвокат зобов'язаний працювати» і «Кожен адвокат, який працює, зобов'язаний носити червоний капелюх», формалізований у (2.12)[32]:

$$\begin{aligned} & \forall x ( \exists e_{11} [(Rexist\ e_{11}) \wedge (\text{адвокат}'\ e_{11}\ x)], \exists e_{21} [(Rexist\ e_{21}\ x) \wedge (\text{працювати}'\ e_{21}\ x)] ) \\ & \forall x ( \exists e \exists e_{11} \exists e_{21} [(Rexist\ e) \wedge (\text{and}'\ e\ e_{11}\ e_{21}) \wedge (\text{адвокат}'\ e_{11}\ x) \wedge (\text{працювати}' \\ & \text{Rexist}\ e_{21}\ x)], \exists e_{31} [(Rexist\ e_{31}) \wedge (\text{носитиЧервонийКапелюх}'\ e_{31}\ x)] ) \end{aligned} \quad (2.12)$$

у разі, якщо система I/O включає аксіому в (2.11),  $O$  повинна включати (2.13), яка відноситься до «Кожен адвокат зобов'язаний носити червоний капелюх»[32].

$$\begin{aligned} & \forall x ( \exists e_{11} [(Rexist\ e_{11}) \wedge (\text{адвокат}'\ e_{11}\ x)], \exists e_{31} [(Rexist\ e_{31}) \wedge ( \\ & \text{носитиЧервонийКапелюх}'\ e_{31}\ x)] ) \end{aligned} \quad (2.13)$$

## 2.7 Формалізація прикладів

Тепер у нас є всі складові для подання (3.1.a-b). У нормативній системі Вводу/Виводу  $N = (O, P)$  попередня вставлена в  $O$ , а остання вставлена в  $P$ . Формула, яка представляє (1.a), дорівнює[32]:

$$\begin{aligned} & \forall x \forall y ( \exists e_1 \exists e_2 [(Rexist\ e_1) \wedge (\text{адвокат}\ x) \wedge (\text{ДЧ}\ y) \wedge (\text{бажати}'\ e_1\ x\ e_2) \wedge ( \\ & \text{практикувати}'\ e_2\ x) \wedge (\text{в}\ e_2\ y) \wedge \text{diffFrom}(y\ f_1(x))], \exists e_3 [(Rexist\ e_3) \wedge (\text{регіструватись}'\ e_3\ x) \\ & \wedge (\text{в}\ e_3\ f_2(y))] ) \end{aligned} \quad (2.14)$$

Як було розглянуто в розділі 4, предикат 'бажання', а також будь-який інший предикат можуть піддаватися різним юридичним інтерпретаціям, які можуть бути затверджені в базі знань за допомогою механізму, використовуваного в Хоббсе для обробки неприпустимості [32].

Дозвіл (1.b) формалізовано аналогічно, (2.15)

$$\forall_y (\exists_x \exists_{e_1} [(Rexist\ e_1) \wedge (\text{інгредієнт}'\ e_1\ x\ y) \wedge (\text{пекарськийМед}\ x) \wedge (\text{харчовийПродукт}\ y)], \exists_{e_1} [(Rexist\ e_2) \wedge (\text{замінювати}'\ e_2\ T_h\ T_{bh}) \wedge (в\ e_2\ f_3(y))]) \quad (2.15)$$

Зверніть увагу, що змінна  $x$  зустрічається тільки в LHS, тому вона кількісно визначена. Формула (2.15) свідчить: для кожного складового харчового продукту  $y$ , для якого воно «істинно» (в тому сенсі, що воно дійсно існує в поточному світі), той факт, що одним з його інгредієнтів є кондитерський мед, тоді дозволено, то що в нинішньому світі також існує факт, що термін «мед» замінюється терміном «кондитерський мед» у назві продукту  $y$  [32].

## 2.8 Глибокий семантико-синтаксичний аналізатор

Виходячи з вище сказаного, пропонується наступна модель глибокого семантико-синтаксичного аналізатора (ГССА):

- На вхід ГССА дані подаються у вигляді тексту викладеному істотною мовою, що описує в загальному випадку алгоритми взаємодії між суб'єктами та об'єктами і суть цієї взаємодії, в окремому випадку правової інформатики це нормативний правової акт.
- На виході з ГССА ми отримаємо набір правил, які придатні для автоматичного створення машинного коду, а також які представлені у вигляді математичних формул.

Алгоритм роботи ГССА:

- A. Вхідні дані передаються на вхід синтаксичного аналізатора, який є частиною ГССА, і результатом роботи якого, ієрархічна структура даних відображає ставлення універсальної залежності, а також синтаксичні і морфологічні дані про елементи з яких складається вхідний текст.
- B. Отримана структура подається на вхід семантичного аналізатора, який є частиною ГССА, і результатом роботи якого є побудова семантичної мережі, яка відображає взаємозв'язки між суб'єктами і об'єктами, і зберігає суть цих взаємодій.
- C. На вхід логічного аналізатора, подається сигнал про зміну семантичної мережі, після чого логічний аналізатор, на підставі зв'язків семантичної мережі створює, доповнює або

змінює **набір машиночитаних логічних правил**. У разі виявлення додаткових семантичних зв'язків на підставі логічного аналізу, логічний аналізатор доповнює семантичну мережу.

D. Отриманий набір машиночитаних логічних правил може бути представлений у вигляді математичних формул для подальшої обробки або перевірки правильності роботи алгоритмів ГССА.

Для перевірки можливості практичної реалізації ГССА, були вирішені завдання зазначені в п. 1.3

## **РОЗДІЛ 3**

### **СЕМАНТИКО-СИНТАКСИЧНИЙ АНАЛІЗ ЗАКОНОДАВЧИХ ДОКУМЕНТІВ**

#### **3.1 Синтаксичний аналіз законодавчого тексту**

Для створення синтаксичного аналізатора був обраний синтаксичний аналізатор SyntaxNet.

Для роботи синтаксичного парсера SyntaxNet необхідна настройка середі виконання. Налаштування середі виконання, а також налаштування.

##### **3.1.1 Налаштування необхідних технічних та програмних засобів для проведення синтаксичного аналізу.**

Для виконання роботи були обрані наступні технічно-програмні средства:

###### **3.1.1.1 Комп'ютерна техніка**

Віртуальна машина:

- ЦПУ: 4 ядерний Intel Xeon E5-processor 2620v3
- ОПЕРАТИВНА ПАМ'ЯТЬ: 8 GB
- Жорсткий диск: SSD 600 GB
- Операційна система Ubuntu 16.04 з встановленими останніми оновленнями.

###### **3.1.1.2 Програмні засоби**

Система управління базами даних - PostgreSQL 9.5

Програмна платформа - Odoo 11.0

Середовище розробки - PyCharm 2017.2.3

Мова програмування - Python 2.7.

Також були встановлені додатково наступні програмні пакети:

- bazel 0.5.4:
- SWIG
- protobuf (буфер протоколу, з версією, підтримуваної TensorFlow)
- mock (пакет тестування)
- asciitree (конструктор дерева розбору на консолі для демонстрації)
- numpy (пакет для наукових обчислень)
- autograd 1.1.13 (для автоматичної диференціації)
- pygraphviz (дерева розбору).

### 3.1.2 Структура бази даних та інтерфейс користувача

Для введення вхідних даних, обробки даних і візуалізації отриманих результатів у цій роботі була обрана безкоштовна платформа ODOO 11.0, що розповсюджується під ліцензією LGPL-3.

Була створена робоча модель **mo\_analytic\_text**.

Для зберігання даних був створений клас **MoAnalyticTextResult(models.Model)** "**mo.analytic.text.result**" з такими полями:

Таблиця 3.1 - Поля моделі mo.analytic.text.result

Найменування	Тип даних	Опис
sequence	Integer	Порядок документу у загальному списку
parent_id	Many2one ( <code>'mo.analytic.text.result'</code> )	Службове поле (використовується для зберігання даних у вигляді ієрархічного списку)
child_ids	One2many ( <code>'mo.analytic.text.result'</code> )	Службове поле (використовується для зберігання даних у вигляді ієрархічного списку)
parent_left	Integer	Службове поле (використовується для зберігання даних у вигляді ієрархічного списку)

parent_right	Integer	Службове поле (використовується для зберігання даних у вигляді ієрархічного списку)
name	Char	Найменування документа, або розділу
source_text	Text	Вхідний текст для аналізу
result_serial	Text	Серіалізовані дані dict, в якому зберігається результат роботи синтаксичного парсера.
result_diagram	Text	Обчислюване поле, яке використовується для формування даних результуючої діаграми, що відображає результати синтаксичного аналізу.
result_text	Text	Результат роботи парсера в форматі CoNLL-U (див. <a href="http://universaldependencies.org/format.html">http://universaldependencies.org/format.html</a> )

Інтерфейс складається з 2-х рівневого меню (рис. 3.1), форми списку розділів (рис. 3.2), а також форми введення і перегляду інформації і отриманих результатів (рис. 3.3).

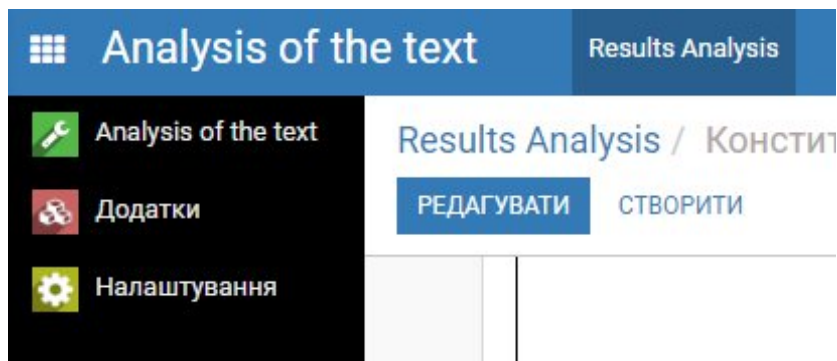


Рисунок 3.4 - Меню інтерфейсу



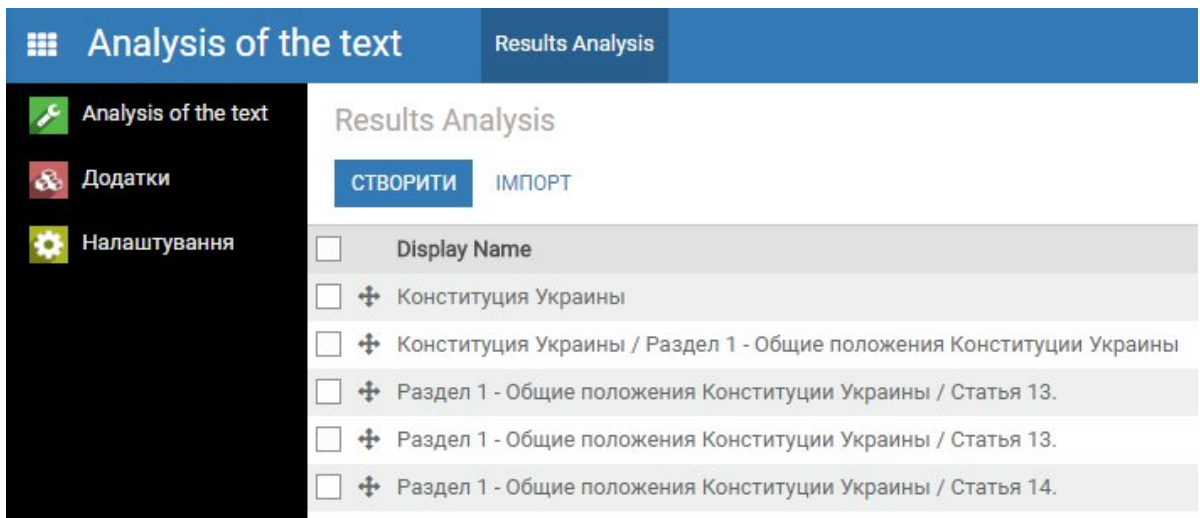


Рисунок 3.5 - Форма списку документів та розділів

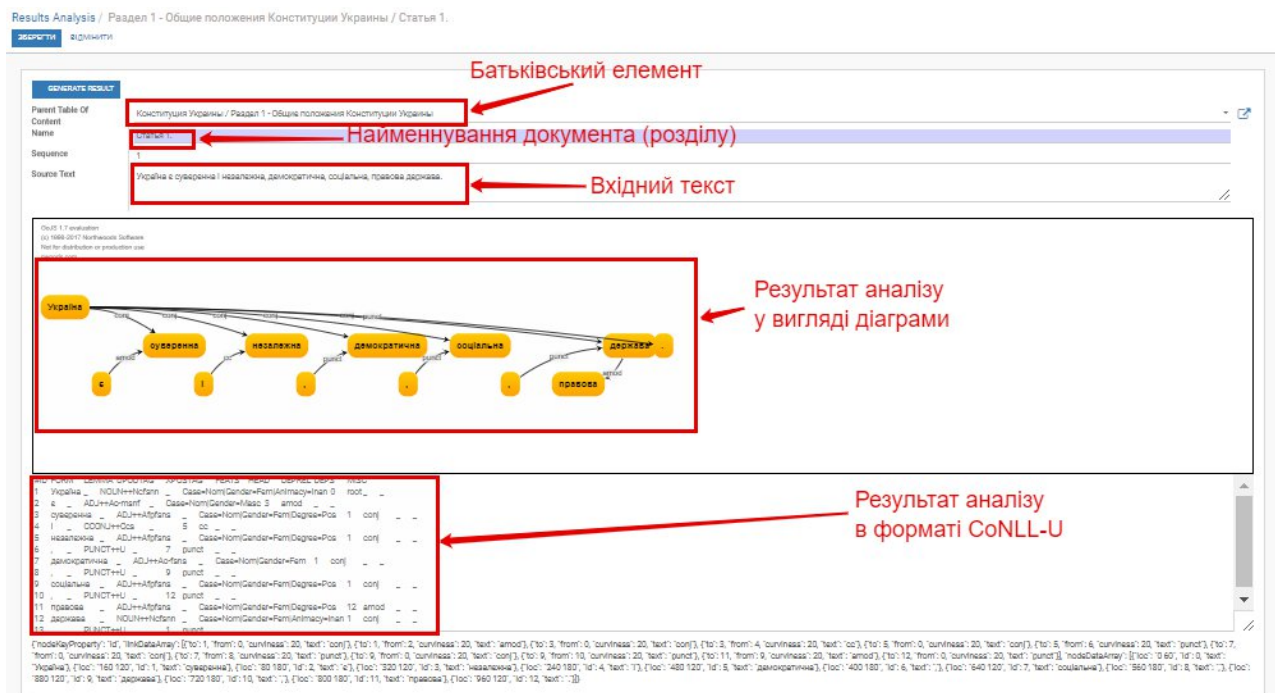


Рисунок 3.6 - Форма введення і просмотра

### 3.1.3 Синтаксичний парсер

Для створення синтаксичного аналізатора був обраний синтаксичний аналізатор SyntaxNet.

Парсер SyntaxNet - це парсер з відкритим вихідним кодом, розроблений компанією Google.

Парсер розроблений на основі штучної нейронної мережі TensorFlow, яка також розроблена компанією Google і так само надається з відкритим вихідним кодом.

SyntaxNet - є ключовим компонентом у багатьох системах NLU (Natural language understanding). З урахуванням речення в якості вхідних даних він поміщає кожне слово з тегом частини мови (POS), який описує синтаксичну функцію слова, і визначає синтаксичні відношення між словами в реченні, представленими в дереві синтаксичного аналізу залежності. Ці синтаксичні відносини безпосередньо пов'язані з основним значенням даного речення.

Існує кілька варіантів установки SyntaxNet:

1. Установка з вільно розповсюджуваних вихідних кодів.
2. Установка в Docker у вигляді web-сервісу.
3. Установка з репозитаріїв python, з допомогою команди pip.

Перший варіант реалізувати не вдалося, дотримуючись офіційної інструкції: <https://github.com/tensorflow/models/tree/master/research/syntaxnet>

Установка завершувалася помилкою, оскільки вихідні коди не проходили тестування за допомогою Bazel.

Спроби знайти рішення, не увінчалися успіхом.

Другий варіант установки був успішним. Було випробувано два варіанти Docker-ов:

1. A small HTTP API for SyntaxNet (<https://github.com/askplatypus/syntaxnet-api>)
2. A simple rest api for Google SyntaxNet (<https://github.com/mitoai/syntaxnet>)

Обидва варіанти реалізують Web-сервер, який запускається в Docker із зазначенням HTTP порту, на який можна надсилати HTTP запити і отримувати результат синтаксичного розбору у вигляді рядка в форматі ConLL-U в першому випадку і у вигляді рядка JSON у другому.

Не дивлячись на всю простоту установки в Docker і простоту використання, істотним недоліком є - робота в запущеному сеансі Docker, в зв'язку з ізоляваністю середовища виконання - важко змінювати і додавати мовні пакети.

В результаті для кінцевої реалізації був обраний третій варіант установки - установка з репозитаріїв python.

### **3.1.4 Проблематика аналізу текстів різноманітних мов**

Наступним кроком в налаштуванні став вибір мовного пакету.

В 3-4 серпня 2017 року у Ванкувері, Канада проходила конференція по Обчислювальному навчанню природним мовам, Computational Natural Language Learning (CoNLL). Конференція CoNLL загальна задача, в якій учасники тренують і тестують свої системи навчання на одних і тих же наборах даних. У 2017 році одна з двох завдань була присвячена вивчають синтаксичних аналізаторів для великої кількості мов, в реальному світі без будь-яких анотацій на основі золотого стандарту.

Корпус що використовувався для підготовки синтаксичних парсерів, містить 5,9 М речень і 90G слів 45 мовами і доступні в форматі CoNLL-U [68]. Розміри для різних мов корпусу перераховані в табл. 3.2.

Таблиця 3.2. - Огляд допоміжних даних: номер слів (М = мільйон, К = тис.) для кожної мови

<b>Мова</b>	<b>Слів (к-сть)</b>
Language Words English (en)	9,441 М
German (de)	6,003 М
Portuguese (pt)	5,900 М
Spanish (es)	5,721 М
French (fr)	5,242 М
Polish (pl)	5,208 М
Indonesian (id)	5,205 М
Japanese (ja)	5,179 М
Italian (it)	5,136 М
Vietnamese (vi)	4,066 М
Turkish (tr)	3,477 М
Russian (ru)	3,201 М
Swedish (sv)	2,932 М
Dutch (nl)	2,914 М
Romanian (ro)	2,776 М
Czech (cs)	2,005 М
Hungarian (hu)	1,624 М
Danish (da)	1,564 М
Chinese (zh)	1,530 М
Norwegian-Bokmal (no)	1,305 М
Persian (fa)	1,120 М
Finnish (fi)	1,008 М
Arabic (ar)	963 М
Catalan (ca)	860 М
Slovak (sk)	811 М
Greek (el)	731 М
Hebrew (he)	615 М
Croatian (hr)	583 М

Ukrainian (uk)	538 M
Korean (ko)	527 M
Slovenian (sl)	522 M
Bulgarian (bg)	370 M
Estonian (et)	328 M
Latvian (lv)	276 M
Galician (gl)	262 M
Latin (la)	244 M
Basque (eu)	155 M
Hindi (hi)	91 M
Norwegian-Nynorsk (no)	76 M
Kazakh (kk)	54 M
Urdu (ur)	46 M
Irish (ga)	24 M
Ancient Greek (grc)	7 M
Uyghur (ug)	3 M
Kurdish (kmr)	3 M
Upper Sorbian (hsb)	2 M
Buryat (bxr)	413 K
North Sami (sme)	331 K
Old Church Slavonic (cu)	28 K
<b>Загалом:</b>	<b>90,669 M</b>

З результатів опублікованих у підсумковому документі конференції видно, що якість обробки текстів на різних мовах залежить від обсягу словника конкретної мови.

Як видно з табл. 3.2, наповнюваність українського словника в більш ніж 6 разів нижче порівняно з російською і більш ніж у 17 разів нижче в порівнянні з англійським.

Ці ж мовні моделі, підготовлені до конференції CoNLL 2017 командою розробників SyntaxNet і доступні в публічному доступі були використані для створення синтаксичного аналізатора, використаного в цій роботі.

Зокрема були випробувані моделі для української, англійської та російської мов.

Результати розбору тексту першої статті Конституції України, у вигляді діаграм, представлені на рисунках:

*"Україна є суверенна і незалежна, демократична, соціальна, правова держава."(с)*

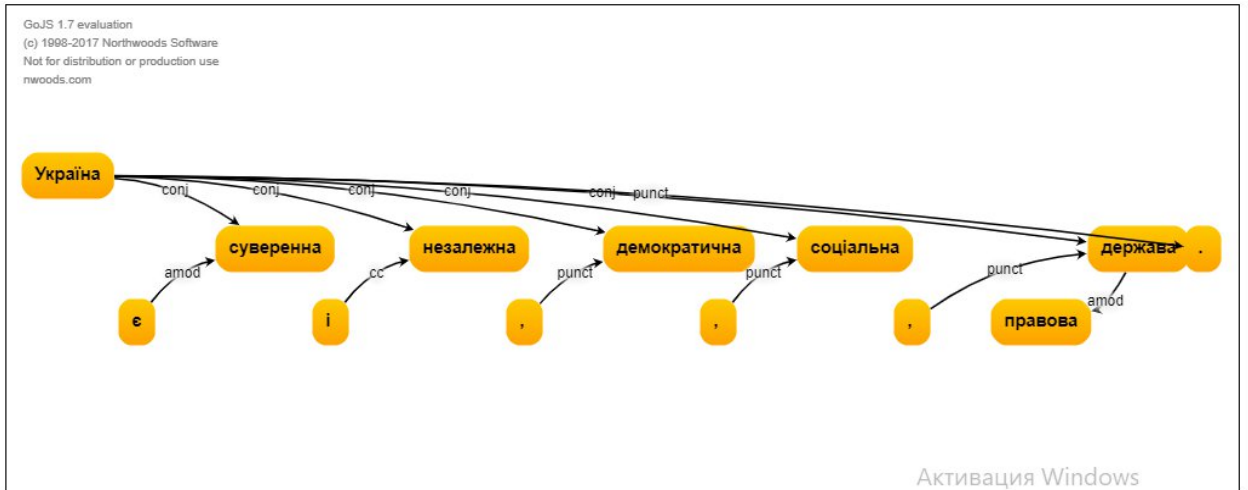


Рисунок 3.7 - Результат синтаксичного аналізу 1 статті українською мовою

*"Ukraine is a sovereign and independent, democratic, social, law-based state."*(c)

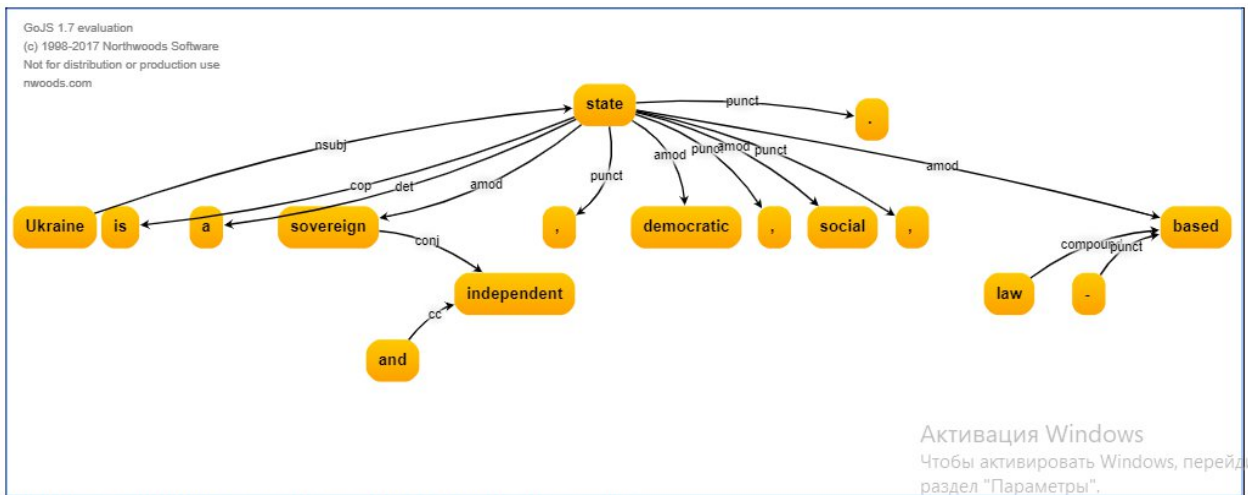


Рисунок 3.8 - Результат синтаксичного аналізу 1 статті англійською мовою

*"Украина является суверенным и независимым, демократическим, социальным, правовым государством."*(c)

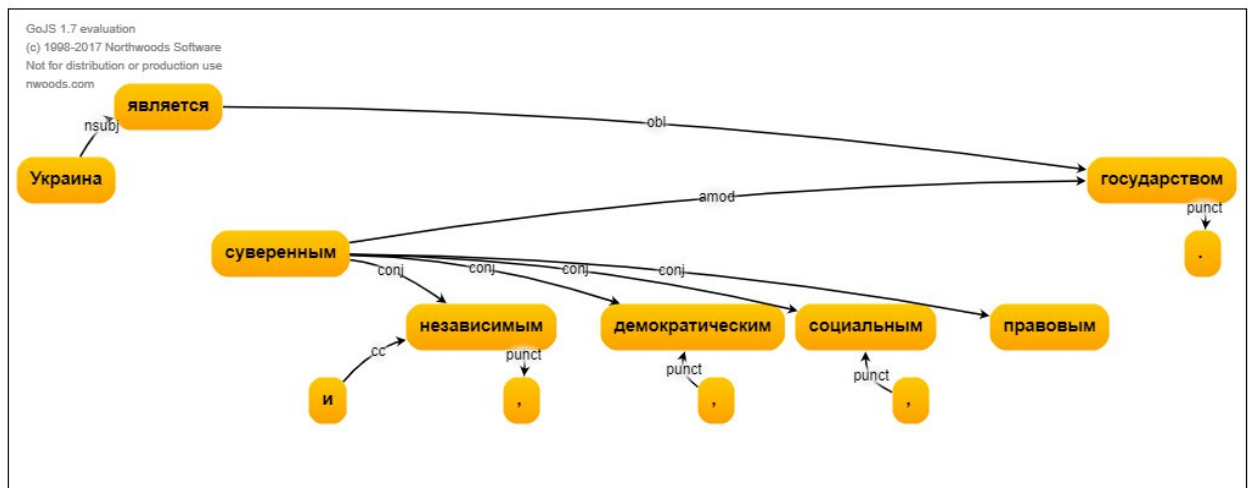


Рисунок 3.9 - Результат синтаксичного аналізу 1 статті російською мовою

Як можливо побачити з результатів аналізу, синтаксичний аналізатор безпомилково впорався за умови використання англійської і російської мовної моделі, показавши при цьому незадовільні результати при використанні української мовної моделі, а саме: при використанні української мовної моделі не вірно відображаються зв'язки між різними частинами речення (чому це так буде пояснено у наступних розділах).

У результаті аналізу результатів був зроблений висновок про неможливість використовувати існуючу українську мовну модель для синтаксичного аналізу українських текстів без додаткового тренування штучної нейронної мережі, використаної в SyntaxNet.

В подальшому, при тестуванні алгоритму семантико-синтаксичного аналізатора в цієї роботі використовувався російська мова, як найбільш близький по синтаксису і морфології до української мови.

### 3.1.5 Аналіз результатів синтаксичного розбору тексту

Результати аналізу представлені в двох видах:

- 1) у вигляді результуючої діаграми;
- 2) у вигляді тексту в форматі CoNLL-U.

Диграмма будується за словами з право на ліво з відступом для кожного слова від попереднього в 80px по горизонталі і з відображенням глибини вкладеності з відступом від попереднього рівня вкладеності в 60px.

Зв'язки між елементами речень вказуються стрілками. Стрілки мають напрямок від меншого індексу частини мови до більшого.

Написи на стрілках показують відносини універсальних залежностей між елементами речень.

Відносини універсальних залежностей описані згідно формату CoNLL-U, а їх опис доступно за адресою: <http://universaldependencies.org/u/dep/index.html>

Результати аналізу відносини універсальних залежностей не показують відношення між елементами різних речень.

В майбутньому такий аналіз відносин можна буде реалізувати з використанням побудованого алгоритму на програмних бібліотеках морфологічного аналізу таких як **rumorphy2** або ж шляхом створення штучної нейронної мережі з подальшою її тренуванням на визначення таких залежностей.

Показовим є порівняння результатів аналізу тексту містить один і той же зміст на різних мовах.

Для прикладу розглянемо результати аналізу 1-ї статті конституції на трьох мовах - українською, російською та англійською:

**Розбір результату синтаксичного аналізу за допомогою української мовної моделі (рис. 3.7):**

*"Україна є суверенна і незалежна, демократична, соціальна, правова держава."(с)*

Головним елементом аналізатор поставив *"Україна"*.

Зв'язки між *"суверенна", "незалежна", "демократична", "соціальна", "держава"* аналізатор позначив тегом **"conj"**, що відповідає кон'юнкту (зв'язок між двома елементами, пов'язаними з координуючим зв'язком, такими як ТА, АБО, тощо), що є не правильним, т. к. ці зв'язки мають відношення до елемента *"держава"*, а не елемента *"Україна"*.

Відповідно єдиною правильною зв'язком (виключаючи знаки припинання), яку зміг визначити аналізатор - є зв'язок між елементами *"держава" - "правова"*, позначену тегом **"amod"**, що відповідає модификатору прикметника(для іменника - будь-яка прилагательна фраза, яка служить для зміни значення іменника).

Таким чином, можна зробити висновок, що існуюча модель української мови для парсера SyntaxNet не є придатною до практичного застосування.

**Розбір результату синтаксичного аналізу за допомогою англійської мовної моделі (рис. 3.8):**

*"Ukraine is a sovereign and independent, democratic, social, law-based state."(с)*

Головним елементом аналізатор поставив *"state"*, це пов'язано з особливостями синтаксису англійської мови, вважаючи на те що зв'язок між важливим елементом *"is"* і головним елементом аналізатор позначив тегом **"cop"**(це відношення функціонального слова, що використовується для зв'язування суб'єкта з неverbальним предикатом), це є вірним.

Причини такої поведінки парсера добре описані в документації стандарту CoNLL-U (<http://universaldependencies.org/u/dep/cop.html>).

Зв'язок з елементом *"Ukraine"* була правильно вибрана тегом **"nsubj"** (номінальний суб'єкт). Решта зв'язку були правильно позначені як **"amod"**, які абсолютно правильно відносяться саме до елемента *"state", "sovereign and independent, democratic, social, law-based"*.

Таким чином, можна зробити висновок, що існуюча модель англійської мови для парсера SyntaxNet придатна для практичного використання.

**Розбір результату синтаксичного аналізу за допомогою російської мовної моделі (рис. 3.9):**

*"Украина является суверенным и независимым, демократическим, социальным, правовым государством."(с)*

Головним елементом аналізатор поставив "являється", що є правильним, оскільки саме цей елемент визначає основні відносини між основними елементами цього тексту.

Зв'язок між елементом "Україна" і головним елементом, також було вірно помічено "nsubj" (суб'єкт).

Зв'язок між елементом "государством" і головним елементом, також було вірно помічено "obj" - об'єкт (об'єкт дієслова є другим найбільш основним аргументом після дієслова суб'єкта).

Зв'язок між елементом "государством" і "суверенным" позначена як "amod", що також є вірним.

На відміну від розбору на англійській мові, в цьому випадку елементи "независимым, демократическим, социальным, правовым" були прив'язані через "conj" до елементу "суверенным". Це на мій погляд більш точно відображає зміст тексту через кон'юнкцію ніж через прикметники як було в результатах розбору англійського тексту.

Таким чином, можна зробити висновок, що існуюча російська мовна модель для парсера SyntaxNet придатна для практичного використання, а зважаючи на близькість російської та української мови по морфології, та синтаксису, російська мовна модель є добрим інструментом для розробки інструментів аналізу текстів на українській мові, до моменту появи якісного українського мовної моделі.

Слід зазначити, що пряме використання російської мовної моделі для аналізу текстів на українській мові не можливо:

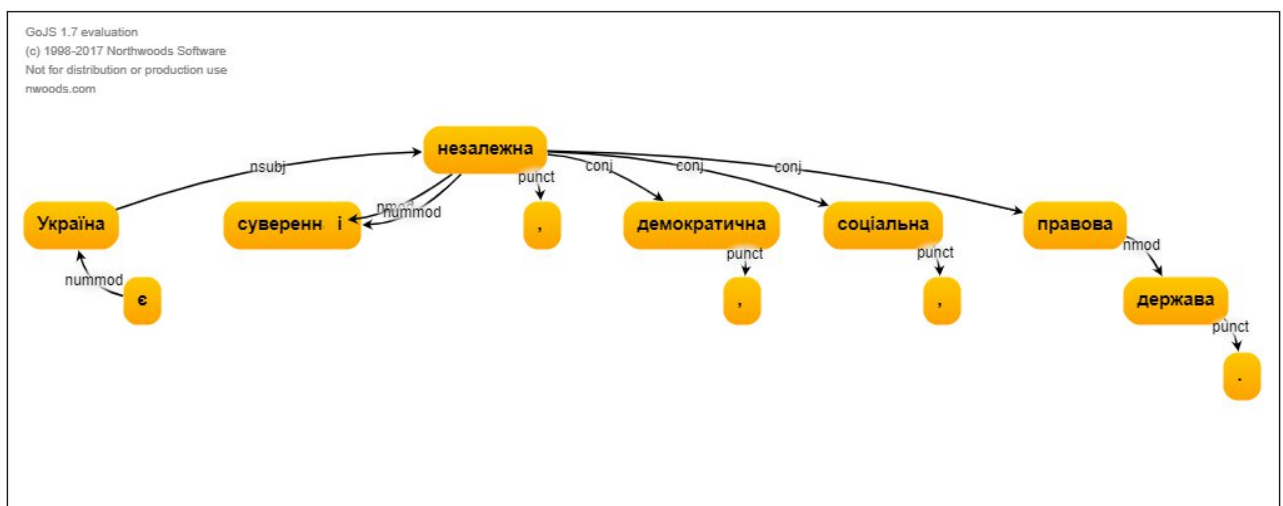


Рисунок 3.10 - Помилковий результат аналізу українського законодавчого тексту за допомогою російською мовної моделі (дуже символічно)



### 3.2 Семантичний аналіз законодавчих документів України

Згідно класичного правознавства (Теорія держави та права <http://studies.in.ua/teorija-derzhavy-ta-prava-lekcii/4396-sklad-zmst-pravovdnosin.html>):

**Склад правовідносин** – сукупність основних елементів правовідносин і спосіб юридичного зв'язку між ними з приводу соціального блага або забезпечення будь-яких інтересів.

Склад правовідносин:

- суб'єкти правовідносин;
- об'єкти правовідносин;
- зміст правовідносин.

**Суб'єкти правовідносин** – індивіди, організації або спільності, які на підставі норм права можуть бути учасниками правовідносин – носіями суб'єктивних прав та юридичних обов'язків.

**Об'єкти правовідносин** – матеріальні, духовні та інші соціальні блага, які служать задоволенню інтересів та потреб громадян та їх організацій, з приводу яких суб'єкти вступають у правовідносини і здійснюють свої суб'єктивні права та юридичні обов'язки.

**Зміст правовідносин** – конкретна поведінка суб'єктів правовідносин та її юридичне закріплення нормами права у вигляді суб'єктивних прав та юридичних обов'язків.

Таким чином, при аналізі юридичних текстів для розуміння їх змісту є обов'язковим розуміння що ж на саме, є об'єктом та суб'єктом правовідносин.

#### 3.2.1 Виявлення суб'єктів та об'єктів правовідносин

За умови якісного синтаксичного аналізу законодавчого тексту процедура виявлення суб'єктів і об'єктів правовідносин не представляє великої проблеми.

Розглянемо простий приклад, перше речення статті 23 Конституції України:

*"Кожна людина має право на вільний розвиток своєї особистості, якщо при цьому не порушуються права і свободи інших людей, та має обов'язки перед суспільством, в якому забезпечується вільний і всебічний розвиток її особистості."*(с)

*"Каждый человек имеет право на свободное развитие своей личности, если при этом не нарушаются права и свободы других людей, и имеет обязанности перед обществом, в котором обеспечивается свободное и всестороннее развитие его личности."*(с)

Виходячи з наведених визначень, в даному випадку суб'єктом правовідносин буде "Кожна людина", а об'єктом правовідносин буде "право на вільний розвиток своєї особистості" і "обов'язки перед суспільством", змістом правовідносин буде "має".

Припустимо:

- головний вузол елементів тексту співпадатиме зі змістом **правовідносин** з урахуванням кон'юкції "та" ("має ... та має");
- суб'єкт правовідносин буде позначений зв'язком **nsubj**;
- об'єкт правовідносин буде позначений зв'язком **obj**.

Результат аналізу можна побачити на рис. 3.11.

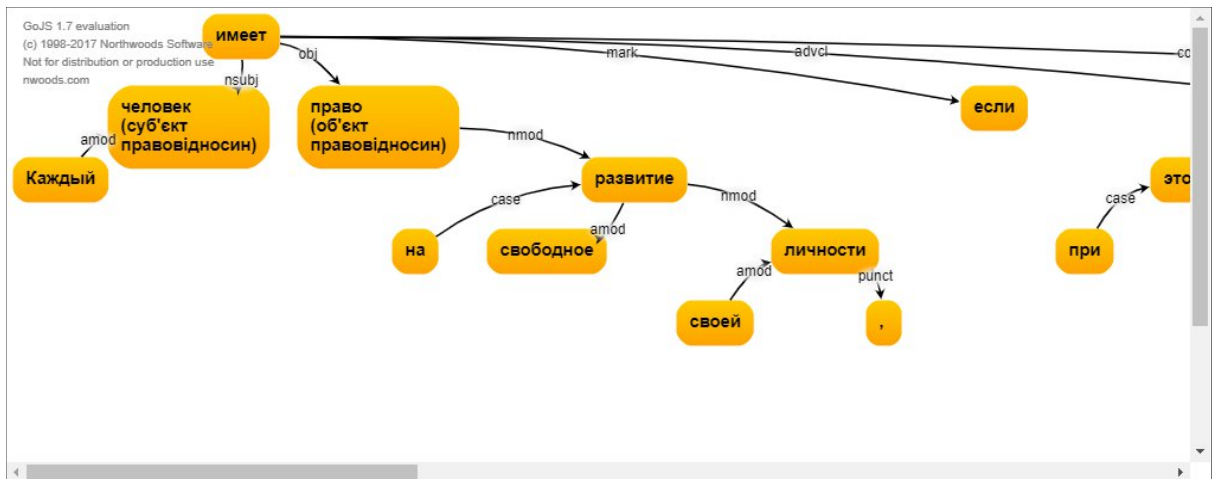


Рисунок 3.11 - Результат семантичного аналізу

Як видно з результату семантичного аналізу, саме так і відбувається.

Це правило спрацьовує для простих випадків, але й для складних, на мій погляд, можливо розробити такі правила відповідності, які дозволять на базі синтаксичного аналізу отримувати склад правовідносин із законодавчих текстів. При цьому варіанти таких правил будуть мати кінцеве число.

В результаті використання результатів такого аналізу, стане можливим будувати семантичні мережі, що відображають склад правовідносин різних частин законодавства, що дозволить визначати набори прав та обов'язків окремих суб'єктів.

### 3.3 Висновки до розділу

За даними що наведені в цьому розділі, практичним шляхом було доведено можливість створення ГССА, виходячи з практичної реалізації його складових частин яка була виконана у цій роботі:

1. Практично продемонстровано роботу синтаксичного аналізатора.
2. Практично продемонстрована робота семантичного аналізатора.

3. Визначена технологія створення логічного аналізатора. На формальному прикладі продемонстровані результати її використання.

Що відповідає задачам цієї магістрської роботи.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Завданням даної дипломної роботи було дослідити бізнес-процеси навчального відділу університету, і як результат було створено модуль управління діяльністю навчального відділу.

Дана робота виконується переважно на персональному обчислювальному обладнанні – комп'ютері. При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів ший, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

#### **4.1 Загальні питання з охорони праці**

##### **4.1.1 Правові та організаційні основи охорони праці**

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

На законодавчому рівні визначено такі пріоритетні напрямки з безпеки праці:

- кожен працівник несе безпосередню відповідальність за порушення зазначених Законом, нормами і правилами вимог;
- напрямки реалізації конституційного права громадян на їх життя і здоров'я в процесі трудової діяльності:
  - пріоритет життя і здоров'я працівників по відношенню до результатів виробничої діяльності підприємства;
  - повна відповідальність роботодавця за створення належних – безпечних і здорових умов праці;
  - соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
  - комплексне розв'язання завдань охорони праці;
  - підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;
  - соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
  - використання економічних методів управління охороною праці, участь держави у фінансуванні заходів щодо охорони праці;
  - використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародної співпраці.

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Обов'язки працівників щодо додержання вимог нормативно-правових актів з охорони

праці (ст. 14), відповідальність робітників всіх категорій за порушення вимог щодо охорони праці (ст. 44) та структура організації/виробництв системи управління охорони праці визначені безпосередньо посадовою інструкцією за посадою інженера ОНС в сфері поводження з відходами № 227-11-2016», та іншими затвердженими власними нормативними актами з питань охорони праці (правилами, нормами, регламентами, положеннями, стандартами, інструкціями та іншими документами, обов'язковими до виконання), тобто тих, що діють на підприємстві/організації, і визначені НПАОП 0.00-6.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві».

#### **4.1.2 Організаційно-технічні заходи з безпеки праці**

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 (НПАОП 0.00-4.12-05).

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 (НАПБ Б.02.005-2003).

Обов'язковими вимогами враховане наступне:

- не слід допускати до роботи осіб, що в установленому порядку не пройшли навчання, інструктаж та перевірку знань з охорони праці, пожежної безпеки та цих Правил.

- на підприємстві/організації, де експлуатуються ЕОМ з ВДТ і ПП, розробляється інструкція з охорони праці відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.98 N 9, зареєстрованого в Міністерстві юстиції України 07.04.98 за N 226/2666 (НПАОП 0.00-4.15-98).

- ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

- перед допуском до самостійної роботи кожен працівник має право на навчання з питань охорони праці і роботодавець зобов'язаний, і проводить таке навчання у вигляді двох інструктажів з питань охорони праці:

1) *вступного*, який проводять працівники служби охорони праці об'єкта господарювання з усіма працівниками, яких приймають на роботу незалежно від їхньої освіти та стажу роботи за програмою, в якій подають загальні питання охорони праці із врахуванням її особливостей на об'єкті господарювання;

2) *первинного*, який проводять керівники структурних підрозділів на місці праці з кожним працівником до початку їхньої роботи на цьому робочому місці.

Проходження працівником цих інструктажів з питань охорони праці підтверджується записами у відповідних журналах обліку інструктажів і скріплюється підписами осіб, які проводили інструктажі та осіб, які отримали інструктажі.

3) *Повторний* (не рідше одного разу в 6 місяців);

4) *Позаплановий* (при зміні правил охорони праці);

5) *Поточний* (проводять з працівниками перед виконанням робіт, на яких оформляється наряд-допуск)

- обов'язкові організаційні заходи перед початком, під час і після завершення роботи повинні включати перевірку (візуально) наявності і справності електрообладнання та його заземлення, а під час виконання роботи вимогу «не залишати без нагляду обладнання, яке працює». Після закінчення роботи - вимагається прибирання робочого місця, відключення всіх електроприладів від електромережі.

Не допускається:

- виконувати обслуговування, ремонт та налагодження ЕОМ з ВДТ і ПП безпосередньо на робочому місці оператора;

- зберігати біля ЕОМ з ВДТ і ПП папір, дискети, інші носії інформації, запасні блоки, деталі тощо, якщо вони не використовуються для поточної роботи;

- відключати захисні пристрої, самочинно проводити зміни у конструкції та складі ЕОМ з ВДТ і ПП або їх технічне налагодження;

- працювати з ВДТ, у яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані тощо;

- працювати з матричним принтером за відсутності вібраційного килимка та зі знятою (піднятою) верхньою кришкою.

## **4.2 Аналіз стану умов праці**

### **4.2.1 Вимоги до приміщень**

Робота над створенням такої системи проходитиме в приміщенні відповідної установи

(компанії, підприємстві тощо). Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером. Геометричні розміри приміщення зазначені в таблиці 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	3
Площа, м <sup>2</sup>	25
Об'єм, м <sup>3</sup>	75

Згідно з [ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам. Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є диван і журнальний стіл а також магнітна дошка. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування. Для забезпечення потрібного рівного освітленості кімната має два вікна та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам. Характеристики робочого місця зазначені в таблиці 4.2.



Приміщення кабінету знаходиться на третьому поверсі п'яти поверхової будівлі і має об'єм 75 м<sup>3</sup>, площу — 25 м<sup>2</sup>. У цьому кабінеті обладнано три місця праці, з яких три укомплектовані ПК.

Таблиця 4.2 - Характеристики робочого місця

<b>Найменування параметра</b>	<b>Фактичне значення</b>	<b>Нормативне значення</b>
Висота робочої поверхні, мм	750	680 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	850	не менше 500
Глибина простору для ніг, мм	750	не менше 650
Висота поверхні сидіння, мм	470	400 500
Ширина сидіння, мм	550	не менше 400
Глибина сидіння, мм	460	не менше 400
Висота поверхні спинки, мм	700	не менше 300
Ширина опорної поверхні спинки, мм	550	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 800

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений

переносним вуглекислотним вогнегасником ВВК-5 .

Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

#### **4.2.3 Навантаження та напруженість процесу праці**

Робота з розробки модулю управління діяльністю навчального відділу за фізичним навантаженням відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК, що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявне психофізіологічні небезпечні та шкідливі фактори:

- а) фізичного перевантаження:
  - статичного;
  - динамічного;
- б) нервово-психічного перевантаження:
  - розумового перенапруження;
  - монотонності праці;
  - перенапруження аналізаторів;
  - емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи наведені в ДСанПіН 3.3.2.007-98

Роботу визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви:

- для розробників програм тривалістю 15 хв через кожну годину роботи.

### 4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### 4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Роботу, пов'язану з електронно-обчислювальними машинами (далі - ЕОМ) з відео дисплейними терміналами (далі - ВДТ), у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і периферійними пристроями (далі - ПП), виконують із забезпеченням виконання НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Робочі місця мають відповідати вимогам цих Правил та Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 (ДСанПіН 3.3.2-007-98). І це передбачає, що визначена виробнича діяльність пов'язана з наявністю певної кількості небезпечних та/або шкідливих виробничих факторів.

Основними робочими характеристиками персонального комп'ютера є наступні:

- робоча напруга  $U=+220\text{В} \pm 5\%$ ;
- робочий струм  $I=2\text{А}$ ;
- споживана потужність  $P=350\text{ Вт}$ .

За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом яких може бути принтер, сканер та ін.), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

#### 4.3.2 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням

розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування. Небезпека загоряння пов'язана з особливістю комп'ютерів - із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100 °С). При проходженні електричного струму по провідниках і деталях виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди). Також ймовірна небезпека внаслідок перевантаження напруги, розрядки зарядів статичної електрики, пошкодження обладнання та електропроводки. Електростатичний розряд виникає під час тертя двох ізольованих матеріалів. Розряд статичної електрики може виникнути під час роботи вентилятора або комп'ютера. Кабельні лінії є найбільш пожежонебезпечними місцем. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання у вигляді електричних іскор і дуг, розгалуженість і недоступність роблять кабельні лінії місцем найбільш ймовірного виникнення і розвитку пожежі. Для зниження займистості і здатності поширювати полум'я кабелі покривають вогнезахисними покриттями. Проектом передбачено прокладати проводку: приховано, під знімною підлогою розділяючи негорючими діафрагмами, в малодоступних місцях.

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними. Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу. Дане приміщення оснащено системою автоматичної пожежної сигналізації, має 1 вогнегасник ВП-5 із зарядом вогнегасної речовини 8-12 кг, відповідно до вимог чинного законодавства України. Проходи до засобів пожежогасіння вільні, не захаращуються та у разі потреби забезпечувати евакуацію всіх людей, які перебувають у приміщенні через один евакуаційний вихід з дверима на шляху евакуації, що відчиняється в напрямку виходу з будівлі від робочого місця. В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі,
- 2) системою протипожежного захисту,
- 3) організаційно-технічними заходами.

Згідно НАПБ Б.03.002-2007 таке приміщення, площею 25 м<sup>2</sup>, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння. Відповідно до норм первинних засобів пожежогасіння пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт.;
- повсть 1 1 м<sup>2</sup>, кошму 2×1,5 м<sup>2</sup> або азбестове полотно 2×2 м<sup>2</sup> в кількості 1 шт.

Виникнення пожежі можливе, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Вірогідність пожежної небезпеки приймається значною, якщо ймовірна взаємодія цих трьох чинників. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, підлоги, двері, ізоляція силових, сигнальних кабелів і т.д.

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) поліамід – матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С,
- 2) полівінілхлорид – ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С,
- 3) склотекстоліт ДЦ – матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання,
- 4) пластикат кабельний №.489 – матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1,
- 5) деревина – будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Для відводу теплоти від ЕОМ діє потужна система кондиціонування. Тому кисень, як окиснювач процесів горіння, є в будь-якій точці приміщень ВЦ.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до [НАПБ Б.03.002-2007] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймісті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Потенційними джерелами запалювання можуть бути:

- 1) іскри і дуги короткого замикання;
- 2) електрична іскра при замиканні і розмиканні ланцюгів;
- 3) перегіви від тривалого перевантаження,
- 4) відкритий вогонь і продукти горіння,
- 5) наявність речовин, нагрітих вище за температуру самозаймання,
- 6) розрядна статична електрика.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол. (ГОСТ 12.1.044-89).

### 4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється -

заземлюючий провідник.

#### 4.4 Гігієнічні вимоги до параметрів виробничого середовища

##### 4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Ia. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»] і наведені в таблиці 4.3:

Таблиця 4.3 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С <sup>0</sup>	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [ДСН 3.3.6.042-99]. Рівні позитивних і негативних іонів у повітрі мають відповідати [ДСН 3.3.6.042-99]. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання. Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату. Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

##### 4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працівника на ПЕОМ.

Робота на ПЕОМ може здійснюватися за таких видах освітлення:

- загальному штучному освітленні, коли відео монітори розташовуються по периметру приміщення або при центральному розташуванні робочих місць у два ряди по довжині кімнати з екранами, звернені в протилежні сторони;

- суміщене освітлення (природне + штучне) тільки при одному і трьох рядном розташуванні робочих місць, коли екран і поверхню робочого столу знаходяться перпендикулярно світла несучій стіні. При цьому штучне освітлення буде виконане стельовими або підвісними люмінесцентними світильниками, рівномірно розміщеними по стелі рядами паралельно світловим прорізам так, щоб екран відео монітора знаходився в зоні захисного кута світильника, і його проекції не доводилися на екран. Працюючі на ПЕОМ не повинні бачити відображення світильників на екрані. Застосовувати місцеве освітлення при роботі на ПЕОМ не рекомендується.

Природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 - 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працює на ПЕОМ. Оптимальна відстань очей до екрана відео монітора повинна становити 60-70 см, допустиме не менше 50 см. Розглядати інформацію ближче 50 см не рекомендується.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IVв розряду точності (середня точність). При



цьому нормована освітленість на робочому місці ( $E_n$ ) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає СНіП 11-4-79 [62]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє СНіП і для даного приміщення в світлий час доби достатньо природного освітлення.

*Розрахунок освітлення.*

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2 + b^2} \cdot S_b = (1/8 \div 1/10) \cdot S_n \quad (4.1)$$

де  $S_b$  – площа віконних прорізів, м<sup>2</sup>;

$S_n$  – площа підлоги, м<sup>2</sup>.

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2$$

$$S_{\text{вік}} = 1/8 \cdot 25 = 3,125 \text{ м}^2$$

Приймаємо 2 вікна площею  $S = 1,6 \text{ м}^2$  кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $N$  виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 25 \text{ м}^2$ ;

$Z$  – поправочний коефіцієнт світильника (для стандартних світильників  $Z = 1.1 - 1.3$ )  
приймаємо рівним 1,1;

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5400лм.

Підставивши числові значення у формулу (4.1), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,64$$

Приймаємо освітлювальну установку, яка складається з 3-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

Потужність електроосвітлювальної установки з урахуванням місцевого освітлення визначається за формулою:

$$N_{\text{Вт}} = \frac{n \cdot W + (0,1 \div 0,2) \cdot n \cdot W}{1000}, \quad (4.3)$$

де  $n$  – розрахункова кількість ламп для освітлення даного приміщення;

$W$  – потужність однієї лампи, Вт;

$(0,1 \div 0,2)$  – додаткова потужність для ламп місцевого освітлення, Вт

$$N_{\text{Вт}} = \frac{3 \cdot 160 + 0,2 \cdot 3 \cdot 160}{1000} = 0,576$$

#### 4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів (зумовлений роботою системних блоків, клавіатури, а також зовнішніми чинниками), коливається у межах 50–65 дБА (ДСН 3.3.6.037-99). Шум такої інтенсивності на тлі високого ступеня напруженості праці негативно впливає на функціональний стан користувачів. Тому на практиці рекомендують знижувати фактичний рівень шуму у приміщеннях, де створюють комп'ютерні програми, виконують теоретичні та творчі роботи, проводять навчання до 40 дБА, а в приміщеннях, де виконують роботу, що потребує зосередженості, — до 55 дБА. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

#### 4.4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти). Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП (30 м<sup>3</sup> на годину на одного працюючого).

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному

провітрюванні.

#### **4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

*1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:*

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:
  - а) якщо об'єм приміщення 20 м<sup>3</sup>, то потрібно подати не менш як 30 м<sup>3</sup>/год повітря;
  - б) якщо об'єм приміщення у межах від 20 до 40 м<sup>3</sup>, то потрібно подати не менш як 20 м<sup>3</sup>/год повітря;

*2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:*

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;
- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин

або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

#### **Вимоги безпеки при надзвичайних ситуаціях:**

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток. При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво. У випадку виникнення пожежі негайно попередити про це пожежну частину та керівництво, виконати усі можливі заходи по евакуації людей з приміщення і розпочати гасіння пожежі первинними засобами пожежогасіння.

2) При замиканні, перевантаженні електричного струму на електричному обладнанні, внаслідок ураження грозової блискавки та ймовірної небезпеки ураженням електричним струмом, приймають наступне:

- попередження замикання здійснюється правильним вибором, монтажем експлуатації мереж;

- застосування захисту схем у вигляді швидкодіючих реле, а також вимикачів, плавких запобіжників, автоматичних вимикачів.

а) У випадку дотику до корпусу та інших струмоведучих частин електроустановки, що опинилися під напругою використовують захисне заземлення - зниження до безпечних значень напруги дотику і кроку, обумовлених замиканням на корпус та ін. Це досягається шляхом, зменшення потенціалу заземленого обладнання (за рахунок підйому потенціалу підстави, на якому стоїть людина, до значення, близького до значення потенціалу заземленого обладнання) та відключення від загальної електромережі ураженого обладнання.

б) У випадку замикання фази на корпус, зниження ізоляції мережі нижче визначеної межі і, нарешті, в разі дотику людини безпосередньо до частини, що знаходиться під напругою. Основними елементами пристрою захисного відключення є прилад захисного відключення і автоматичний вимикач.

*Прилад захисного відключення* - сукупність окремих елементів, які приймають вхідну величину, реагує на її зміни і при заданому значенні дають сигнал на її відключення вимикача:

- датчику - вхідна ланка пристрою, що сприймають впливу ззовні і здійснюють перетворення цього впливу в відповідний сигнал;

- підсилювача, призначений для посилення сигналу датчика, якщо він виявляється

недостатньо потужним;

- ланцюгів контролю, службовці періодичної перевірки справності захисного відключення;

- допоміжних елементів - сигнальні лампи і вимірювальні прилади, що характеризують стан електроустановки.

*Автоматичний вимикач* - апарат, призначений для включення і вимикання від ланцюгів під навантаженням і при коротких замиканнях. Він повинен включати ланцюг автоматично при надходженні сигналу від приладу захисного відключення.

Також застосовують різні **електричні захисні засоби від ураження струмом**:

*а) Ізолюючі* - ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.

*б) Основні* - володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою. До них відносяться: в електроустановках до 1000 Вт - діелектричної рукавички, ізолюючі штанги, ізолюючі і електровимірювальні кліщі і т.д .; понад 1000Вт - ізолюючі штанги, і електровимірювальні кліщі, а також кошти для ремонтних робіт під напругою понад 1000Вт.

*в) Запобіжні* - володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, при чому при використанні основних захисних засобів достатньо застосування одного запобіжного захисного засобу. До запобіжних відносяться засоби в електроустановках до 1000Вт - діелектричні калоші килимки, а також ізолюючі підставки.

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

**Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).**

Загальний опір захисного заземлення визначається за формулою:

$$R_{ззп} = \frac{R_з \cdot R_n}{R_n \cdot n \cdot \eta_з + R_з \cdot \eta_n}, \quad (4.4)$$

де  $R_з$  - опір заземлення, якими когут бать труби, опори, кути і т.п., Ом;

$R_n$  - опір опори, яке з'єднує заземлювачі, Ом;

$n$  - кількість заземлювачів;

$\eta_3$  - коефіцієнт екранування заземлювача; приймається в межах  $0,2 \div 0,9$ ;  $\eta_3 = 0,7$

$\eta_{ш}$  - коефіцієнт екранування сполучної стійки; приймається в межах  $0,1 \div 0,7$ ;  $\eta_{ш} = 0,5$ ;

Опір заземлення визначається за формулою:

$$R_3 = \frac{\rho}{2\pi \cdot l} \cdot \left( \ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.5)$$

де  $\rho$  - питомий опір ґрунту, залежить від типу ґрунту, Ом м; для піску -  $400 \div 700$  Ом • м;

$\rho = 400$  Ом • м;

$l$  - довжина заземлювача, м; для труб - 2-3 м;  $l = 3$  м;

$d$  - діаметр заземлювача, м; для труб - 0,03-0,05 м;  $d = 0,05$  м;

$t$  - відстань від середини забитого в ґрунт заземлювача до рівня землі, м;  $t = 2$  м.

$$R_3 = \frac{400}{2 \cdot 3,14 \cdot 3} \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 2 + 3}{4 \cdot 2 - 3} \right) = 110, \text{ Ом}$$

Опір шваби, що з'єднує заземлювачі, визначається за формулою:

$$R_{ш} = \frac{\rho}{2\pi \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot t^1}, \quad (4.6)$$

де  $L$  - довжина шваби, що з'єднує заземлювачі, м; приблизно дорівнює периметру будівлі;

$P_{зд.} = 42 \cdot 2 + 38 \cdot 2 = 160$  м;  $L = 160$  м;

$b$  - ширина шваби, м;  $b = 0,03$  м;

$t_1$  - глибина заземлення від рівня землі, м;  $t_1 = 0,5$  м.

$$R_{ш} = \frac{400}{2 \cdot 3,14 \cdot 160} \cdot \ln \frac{2 \cdot 160^2}{0,03 \cdot 0,5} = 5,99, \text{ Ом}$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3}, \quad (4.7)$$

де  $4$  - допустиме загальний опір, Ом;

$2$  - коефіцієнт сезонності.

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3}$$

Визначаємо загальний опір захисного заземлення:

$$R_{ззп} = \frac{110 \cdot 5,99}{5,99 \cdot 79 \cdot 0,7 + 110 \cdot 0,5} = 1,7 \text{ Ом}$$

Висновок: дане захисне заземлення буде забезпечувати електробезпека будівлі, так як виконується умова -  $R_{ззп} < 4$  Ом.

3) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

Важлива увага слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинен накопичуватися сміття, непотрібний папір, мотлох та ін. Речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі.

## **4.6 Охорона навколишнього природного середовища**

### **4.6.1 Загальні дані з охорони навколишнього природного середовища**

Діяльність за темою дипломної роботи управління діяльністю навчального відділу в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря

робочої зони.

В процесі діяльності за системою автоматизації управління діяльністю навчального відділу виникають процеси поводження з відходами. Види відходів, що утворюються в процесі:

Відпрацьовані люмінесцентні лампи - I клас небезпеки

Відпрацьовані вогнегасники - IV клас небезпеки

Макулатура - IV клас небезпеки

Матеріали текстильні вторинні (спецодяг, х/б, шерстяні) - IV клас небезпеки

Відпрацьовані фільтрувальні засоби індивід. захисту (респіратори, протигази) - IV клас небезпеки

Побутові відходи - IV клас небезпеки

#### 4.6.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог Державних санітарних правил і норм ДСанПіН 2.2.7.029.

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення. Зазначені для зберігання відходів місця чи об'єкти повинні використовуватися лише для заявлених відходів.

Не допускається зберігання відходів у невстановлених схемою місцях, а також перевищення норм тимчасового зберігання відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери).

У міру наповнення тару з відходами закривають герметично сталевий кришкою;

- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автотранспортом перевантажують у самоскид і доставляють на місце утилізації або захоронення;

- В разі тимчасового зберігання відходів у стаціонарних складах або промислових приміщеннях повинні бути забезпечені санітарно-гігієнічними етичними вимогами до повітря робочої зони згідно з ГОСТ 12.1.005.

Не допускається змішування відходів різних видів і класів небезпеки з будівельними і побутовими відходами, відходами дерев'яної, металевої, синтетичної тари, відходами текстильних матеріалів (старий спецодяг, ганчірки) і інш.

Особливий контроль наділяється збору і зберіганню відпрацьованих ртутьмісних ламп



(енергоощадних) як відходам I класу небезпеки, що збираються і обов'язково передаються на утилізацію підприємствам, що мають ліцензію на поводження з такими небезпечними відходами.

Всі відходи, що утворюються в процесі діяльності/роботи, підлягають обліку.

Вимоги безпеки при поводженні з відходами:

Під час роботи з відходами (прибирання виробничих приміщень, збір і сортування, навантаження, транспортування, розвантаження та ін.) працівники та обслуговуючий персонал підприємства повинні бути забезпечені засобами індивідуального захисту та дотримуватися вимог інструкцій з охорони праці, що діють на підприємстві.

Наведено перелік деяких відходів, які передаються на утилізацію організаціям, які мають ліцензію на поводження з відходами як вторинної сировини:

- Макулатура;
- Матеріали текстильні вторинні;
- Відпрацьовані фільтрувальні засоби індивідуального захисту
- Відпрацьовані вогнегасники
- Матеріали пакувальні вторинні

Відвантаження таких відходів здійснюється відповідно до договору (контракту)

Побутові та будівельні відходи вивозяться на полігон твердих побутових відходів міста, також відповідно до договору з комунальним дорожньо-експлуатаційним управлінням.

Особи, винні в порушенні встановленого порядку поводження з відходами (порушення правил обліку відходів, самовільне складування і видалення відходів, передача відходів в інші підприємства/організації з порушенням встановлених правил), згідно законодавства несуть дисциплінарну, адміністративну або кримінальну відповідальність.

#### **4.6.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі**

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів. Відомості про місце утворення та місце розташування відходів зазначаються на «План схемі місці розміщення відходів організації / виробництва» та наводяться у таблиці 4.4, а Відомості про склад і властивості відходів, що утворюються, а також ступінь їх небезпечності для навколишнього природного середовища та здоров'я людини у таблиці 4.5 (*в таблицях приведена більшість відходів, що характерна для організацій, основна діяльність, яких пов'язана з роботою на ПЕОМ - треба вибрати потрібне*).

## Відомості про місце утворення та місце розташування відходів

Таблиця 4.4

№ з/п	Код та найменування відходів за ДК -005-96	Технологічний процес або виробництво, де утворюються відходи / клас небезпеки	Місце розташування відходу, тара та її кількість, місткість, розміри у разі наявності майданчиків розташування відходів необхідно зазначити тип покриття та наявність даху)	№ на схемі (додається масштабна схема місць розміщення відходів)
1	2	3	4	5
1	7710.3.1.26 Лампи люмінесцентні, та відходи, які містять ртуть, інші зіпсовані або відпрацьовані (Відпрацьовані ртутьвмісні люмінесцентні лампи)	1	буд.17, в приміщені кладової S=20м <sup>2</sup> , в кількість 5 од.	1701-ТХ
2	7720.3.1.01 Відходи комунальні (міські) змішані, у т.ч. сміття з урн (Побутові відходи)	4	зовнішній майданчик зберігання побутових відходів біля буд .17 S=5м <sup>2</sup> V= 2,08м <sup>3</sup> - 2од.	1701-ТХ
3	7710.3.1.01 Макулатура паперова та картонна (Макулатура)		буд .14 3 поверх S =0,5 м. <sup>2</sup>	1701-ТХ
4	7730.3.1.07 Одяг захисний зіпсований, відпрацьований чи забруднений (Матеріали текстильні вторинні) спецодяг (х/б, шерстяні)	4	буд .17 1 поверх кладова S=2м <sup>2</sup> V= 0,2м <sup>3</sup> - 1 од.	1701-ТХ
5	7730.3.1.02 Матеріали пакувальні пластмасові зіпсовані, відпрацьовані чи забруднені (Матеріали пакувальні забруднені)	4	буд.17, контейнер V=0,9м <sup>3</sup> (3 од.)	1701-ТХ
6	8530.2.9.03 Засоби захисту від хімічних або бактеріальних аерозолів зіпсовані, або відпрацьовані (Відпрацьовані фільтрувальні засоби індивідуального захисту)	4	буд .17- 3 поверх кладова S=2м <sup>2</sup> V= 0,64м <sup>3</sup> - 1од.	1701-ТХ
7	Змінні носії інформації	4	буд. 17, кім. 12 V=0,0005 м <sup>3</sup>	1701-ТХ

Таблиця 4.5 Відомості про склад і властивості відходів, що утворюються, а також ступінь їх небезпечності для навколишнього природного середовища та здоров'я людини

№ п / п	Назва відходу	Клас небезпечності	Хімічний та морфологічний склад	Фізико-хімічні властивості	Негативний вплив на навколишнє середовище та здоров'я людини
1	2	3	4	5	6
1	Відпрацьовані люмінесцентні лампи	I	<b>Ртуть</b> – 0,013 <b>Hg</b>  <b>Скло</b> – 98,787 <b>(Na, K)<sub>2</sub>O 2SiO<sub>2</sub></b>  <b>Алюміній</b> – 1,2 <b>Al</b>	<b>Ртуть</b> – T <sub>кип.</sub> = 356,58С T <sub>плав.</sub> = - <b>38,87С</b>  <b>Скло</b> – T <sub>плав.</sub> = 800С  <b>Алюміній</b> – T <sub>кип.</sub> = 2348С T <sub>плав.</sub> = 660,1С	<b>Ртуть</b> Являючись потужним кумулятивним отрутою, з можливою канцерогенною і мутагенною дією. Процеси самоочищення водою порушують концентрація ртуті понад 0,018 мг/л, порогова концентрація ртуті за впливом на санітарний режим водою 0,01 мг/л. Наприкінці концентрація понад 0,03 є токсичною практично для всіх видів водних організмів. Надзвичайно токсична при попаданні з питною водою для теплокровних організмів, надходження ртуті з питною водою в кількості 75,0 – 300,0 мг/сут є смертельним. При отруєнні парами спостерігається слабкість, головний біль, біль в шлунку, роздратування почек. Розвивається всього тіла. Виникає стан підвищеної психічної збудливості [41]. Пари ртуті проявляють нейротоксичність, особливо страждають вищі відділи нервової системи [52] <b>Скло</b> Нетоксичні, безпечно в навколишньому середовищу, не шкідлива в нирках і водоймах. Вдихання скляного пилу призводить до силікоз в зв'язку з високим вмістом сполук кремнію. Шкідливої дії не робить, але є небезпека механічних пошкоджень (порізи, травми). <b>Алюміній</b> Токсичний для водної біоти, теплокровних тварин і людей, в концентрації > 1 мг/л чинить негативний вплив на зростання с/г культур. У концентрації > 1 мг/л гальмує зростання мікрофлори водою і стримує процеси самоочищення водою. Рівень токсичності визначається формою, в якій знаходиться елемент. Впливає на обмін речовин і функції нервової системи [52].

Продовження табл. 4.5

1	2	3	4	5	6
2	Макулатура	IV	<p><b>Цинк</b> – 0,000053 – 0,000056 <b>Zn</b></p> <p><b>Свинець</b> – 0,000049 – 0,000051 <b>Pb</b></p> <p><b>Хром</b> – 0,000051 – 0,000054 <b>Cr</b></p> <p><b>Мідь</b> – 0,000033 – 0,000035 <b>Cu</b></p> <p><b>Целюлоза</b> – 97,299814 – 96,999804 <b>(C<sub>6</sub>H<sub>10</sub>O<sub>5</sub>)<sub>n</sub></b></p> <p><b>Вода</b> – 2,7 – 3,0</p>	<p>Уривки та обрізки з паперових мішків</p> <p><b>Цинк</b> T<sub>кип.</sub> = 913C T<sub>плав.</sub> = 4,19C</p> <p><b>Свинець</b> T<sub>кип.</sub> = 1751C T<sub>плав.</sub> = 327,3C</p> <p><b>Хром</b> T<sub>кип.</sub> = 1890C T<sub>плав.</sub> = 2480C</p> <p><b>Мідь</b> T<sub>кип.</sub> = 2580C T<sub>плав.</sub> = 1083C</p> <p><b>Целюлоза</b> T<sub>возг. с обуглив.</sub> 100C</p>	<p><b>Цинк</b> Малотоксичний для теплокровних тварин при надходженні з їжею. Концентрація в питній воді 11,2 – 26,6 мг/л переноситься без будь-яких ознак інтоксикації. Дуже корисний для флори, один з найважливіших мікроелементів харчування, однак лише в концентрації до 0,2 мг/л. Дуже токсичний для водних організмів, порушуючи процеси самоочищення водойм і стаючи токсичним для іхтіофауни в концентрації 0,15 – 5,0 мг/л. Мутагенна і онкогенна небезпеку [52].</p> <p><b>Свинець</b> У концентрації 2,0 мг/л надає воді металевий присмак. В концентрації 1,90 мг/л згубно діє на дафній, концентрація 0,1 мг/л погіршує процеси самоочищення водойм. Свинець токсичний для рослин в концентрації понад 5,0 мг/кг ґрунту. Помірно токсичний. Вражає центральну і периферичну нервову систему, кістковий мозок і кров, судини, синтез білка, генетичний апарат клітини [52].</p> <p><b>Хром</b> Міститься в природних водах в концентрації 0,001 – 0,112 мг/л. LK50 Cr (VI) для риб-30,0 – 50,0 мг/л, LK50 Cr (III) для риб – 117,0 мг/л. Низькі концентрації хрому позитивно впливають на ріст рослин, проте полив водою культур з концентрацією хрому 10,0 – 50,0 мг/л гальмує їх розвиток.</p> <p><b>Мідь</b> Дуже токсична як для водних організмів, так і для рослин. У концентрації 0,001 мг/л гальмує розвиток синьо-зелених водоростей, LK50 практично для всіх видів риб становить 0,18 – 1,35 мг/л. У концентрації 0,1 – 0,2 мг/л надає токсичну дію на ріст рослин. Викликає гостре отруєння.</p> <p><b>Целюлоза</b> Нетоксична. Досить легко підвержен біодеструкції лігнін і целюлозоруйнучими бактеріями і деякими класами нижчих грибів. Токсичність визначається за вмістом важких металів, здатних мігрувати з неї в навколишнє середовище.</p>

Продовження табл. 4.5

1	2	3	4	5	6
3	Побутові відходи	IV	<p>Побутові відходи – 100 – 100, в т. ч.:</p> <p>Папір - 30 - 17; [(C<sub>6</sub>H<sub>10</sub>O<sub>5</sub>)<sub>n</sub> – целюлоза]</p> <p>Поліетилен - 20 – 24; (CH<sub>2</sub> – CH<sub>2</sub>)<sub>n</sub></p> <p>Деревина – 5 – 3; [(C<sub>6</sub>H<sub>10</sub>O<sub>5</sub>)<sub>n</sub> – целюлоза, лігнін]</p> <p>Матеріали текстильні – 4 – 3; [(C<sub>6</sub>H<sub>10</sub>O<sub>5</sub>)<sub>n</sub> – целюлоза</p> <p>Мінеральні домішки (пісок, глина) – 4 – 9</p> <p>Харчові відходи – 37 – 44</p>	<p>Целюлоза T<sub>возг. с обуглив.</sub> 100С</p> <p>Поліетилен - T<sub>размяг.</sub> 150С</p> <p>Твердий матеріал рослинного походження. Целюлоза, лігнін T<sub>возг. с обуглив.</sub> 120С</p> <p>Твердий матеріал рослинного походження, не розчиняється у воді. Целюлоза T<sub>возг. с обуглив.</sub> 100С</p> <p>Харчові відходи T<sub>биоразл.</sub> 4° С</p>	<p><b>Целюлоза</b> Нетоксична. Досить легко піддавався біодеструкції лігнін- і целюлозоруйнучими бактеріями і деякими класами нижших грибів. У зв'язку з нетоксичністю LD50 для тваринах не установлена. Токсичність визначається за вмістом важких металів, здатних мігрувати з неї в навколишнє середовище.</p> <p><b>Поліетилен</b> Нетоксичний для всіх видів флори і фауни в зв'язку з дуже високою біологічною інертністю. Нерозчинний у водних середовищах і не впливає на санітарний режим водойм. Використання його не вимагає запобіжних заходів. Отруєння можливі при виробництві та переробці плівки, в результаті виділення окису вуглецю, альдегідів, органічних кислот [43]</p> <p><b>Деревина</b> Нетоксична. Досить легко піддається біодеструкції лігнін і целюлозоруйнучими бактеріями і деякими класами нижчих грибів. У зв'язку з нетоксичністю LD50 для тварин не встановлена. Деревина нетоксична при використанні. Але дія деревного пилу при рубці і переробці деревини викликає захворювання дихальних шляхів і шкіри.</p> <p><b>Текстильне волокно</b> Нетоксична в зв'язку з біогенним походженням, проте для біодеструкції необхідна наявність вологи. Нетоксична при використанні. Токсична дія виникає (як результат механічної дії наслідок пилу) при виробництві тканив і при переробці вторинних матеріалів; слабкий алерген [59].</p>

#### 4.7 Висновки до розділу

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

## ВИСНОВКИ

Як було зазначено, метою магістерської роботи є створення ефективного алгоритму для смислового аналізу законодавчих текстів українського законодавства, за допомогою існуючих програмно-алгоритмічних засобів семантико-синтаксичного аналізу та доказ можливість його практичного використання.

У результаті роботи була запропонована теоретична модель глибокого семантико-синтаксичного аналізатора (ГССА), описано загальні принципи його роботи, визначено його складові елементи, а також описано його загальний алгоритм роботи.

Також була практичним шляхом було доведено можливість створення ГССА, виходячи з практичної реалізації його складових частин яка була виконана у цій роботі:

1. Практично продемонстровано роботу синтаксичного аналізатора.
2. Практично продемонстрована робота семантичного аналізатора.
3. Визначена технологія і показана на формальному прикладі робота логічного аналізатора.

Так само були отримані негативні результати, а саме, відсутність якісної української мовної моделі синтаксичного парсера призводить до практичної неможливості аналізу текстів українського законодавства. При цьому теоретична така можливість була доведена в цій роботі. Слід також зазначити, що в умовах існування інших мовних моделей, близьких по синтаксису і морфології української, завдання створення якісної української мовної моделі представляється відносно легко здійсненним.

Використання ГССА підвищить ефективність аналізу законодавчих текстів законодавства на слов'янських мовах, дозволить створити інструменти, які можуть допомогти створювати прозоре законодавство, допоможуть в пошуку протиріч у законодавчих нормах, дозволять уникати матеріальних втрат у зв'язку з правовою неосвіченістю та протидіяти корупції в органах судової та державної влади.

Межі практичного застосування ГССА, у разі його практичної реалізації, не обмежується областю правової інформатики. ГССА може широко використовуватися і в інших областях інформатики, наприклад:

- створення морально-етичних запобіжників систем штучного інтелекту;
- створення машинного коду шляхом опису його природною мовою;
- створення алгоритмів смарт-контрактів без знання програмування;
- створення інтелектуальних систем віртуальних помічників і т. д.

Таким чином мета цієї магістерської роботи була досягнута.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. "D2.1 Collection of state-of-the-art NLP tools for processing of legal text" Contributors: UL, UNITO, UNIBO, INRIA, DATA61 (2017)
2. L. Robaldo, L. Lesmo, D. Radicioni: Compiling Regular Expressions to Extract Legal Modifications, In proc. of 25th International Conference on Legal Knowledge and Information Systems (JURIX2012). Amsterdam, 2012.
3. G. Boella, L. Di Caro, A. Ruggeri, L. Robaldo. Learning from syntax generalizations for automatic semantic annotation. *Journal of Intelligent Information Systems* 43 (2), 231-246
4. L. Lesmo. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June 2007.
5. Deerwester, Scott, et al. "Indexing by latent semantic analysis." *Journal of the American society for information science* 41.6 (1990): 391.
6. Hofmann, Thomas. "Probabilistic latent semantic indexing." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.
7. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
8. Huang, W.C. "A Patent Document Category System by Using Stanford Parser a Rough Set Theory" in *Managing access to the internet in public libraries in the UK: the findings of the MAIPLE project*, Spacey, Rachel et al. (eds.), ATINER, 2014.
9. Wyner, Adam, and Wim Peters. "On Rule Extraction from Regulations." *JURIX*. Vol. 11. 2011.
10. Testerink, B. Floris, B.: Demo: Natural Language Processing for Online Fraud Scenario Extraction, demo presented at the 22nd European Conference on Artificial Intelligence (ECAI 2016).
11. Floris Bex, Joeri Peters and Bas Testerink: A.I. for Online Criminal Complaints: From Natural Dialogues to Structured Scenarios, in *prof. of Artificial Intelligence for Justice workshop*, collocated at the 22nd European Conference on Artificial Intelligence (ECAI 2016).
12. Vico Horacio, Calegari D. Software Architecture for Document Anonymization, *Electronic Notes in Theoretical Computer Science (ENTCS)*, Vol. 314, Issue C, June 2015.
13. Schilder, F. Event extraction and temporal reasoning in legal documents. *Proc. of the*



2005 international conference on Annotating, extracting and reasoning about time and events.

14. Vila, L., Yoshino, H.: Time in automated legal reasoning. *Information and Communications Technology Law* 7, 173–197 (1998)
15. Neural Reasoning For Legal Text Understanding (Guido Boella Adebayo Kolawole John, Luigi Di Caro), In *Proceedings of the 29th International Conference on Legal Knowledge and Information Systems (JURIX2016)*, 2016.
16. Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.
17. LR Medsker and LC Jain. *Recurrent neural networks. Design and Applications*, 2001.
18. Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.
19. Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735– 1780, 1997.
20. Schradang, J Nicolas, "Analyzing Domestic Abuse using Natural Language Processing on Social Media Data" (2015). *esis*. Rochester Institute of Technology.
21. Mariolina Eliantonio Marta Ballesteros, Rostane Mehdi and Damir Petrovic. Tools for ensuring implementation and application of eu law and evaluation of their effectiveness, July 2013.
22. Luis Polo Paredes, JM Rodríguez, and Emilio Rubiera Azcona. Promoting government controlled vocabularies for the semantic web: the eurovoc thesaurus and the cpv product classification system. *Semantic Interoperability in the European Digital Library*, page 111, 2008.
23. Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.
24. Mauro Dragoni, Serena Villata, Williams Rizzi and Guido Governatori, Combining NLP Approaches for Rule Extraction from Legal Documents. In *Proceedings of the 1st Workshop on `Mining and REasoning with Legal texts'*, 2016.
25. Wyner, Adam and Bos, Johan and Basile, Valerio and Quaresma, Paulo. "An empirical approach to the semantic representation of laws." *JURIX*. 2012.
26. G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco, "Computing Strong and Weak Permissions in Defeasible Logic," *Journal of Philosophical Logic*, vol. 42, no. 6, pp. 799–829, 2013.
27. G. Governatori and S. Sadiq, "The Journey to Business Process Compliance," in

- Handbook of Research on Business Process Modeling, J. Cardoso and W. M. P. van der Aalst, Eds. Hershey, New York:, 2009, pp. 426–454.
28. G. Governatori, “Representing business contracts in RuleML,” *International Journal of Cooperative Information Systems*, vol. 14, no. 2-3, pp. 181–216, 2005.
  29. Koehn, Philipp. *Europarl: A Parallel Corpus for Statistical Machine Translation*, MT Summit 2005.
  30. C. Bartolini, G. Lenzini and L. Robaldo: *Towards legal compliance by correlating Standards and Laws with a semi-automated methodology*, in *proc. of the 28th Annual Benelux Conference on Artificial Intelligence*. Amsterdam, 2016.
  31. G. Boella, L. Di Caro, L. Humphreys, L. Robaldo, P. Rossi, L. van der Torre: *Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law*, *Artificial Intelligence and Law*, 24 (3).
  32. *Reified Input/Output logic - a position paper* (Livio Robaldo, Xin Sun), In *Proceedings of Workshop on Artificial Intelligence for Justice (AI4J)*, collocated at the 22nd European Conference on Artificial Intelligence (ECAI 2016), The Hague, The Netherlands., 2016.
  33. G. Boella, L. Di Caro, A. Ruggeri, and L. Robaldo, ‘Learning from syntax generalizations for automatic semantic annotation’, *Journal of Intelligent Information Systems*, 43(2), 231–246, (2014).
  34. G. Boella, L. Di Caro, and L. Robaldo, ‘Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines’, in *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pp. 218–225. Springer, (2013).
  35. G. H. von Wright, *An Essay in Deontic Logic and the General Theory of Action*, *Acta Philosophica Fennica*, Fasc. 21, North-Holland, 1968.
  36. J.J. Meyer and R.J. Wieringa, *Deontic Logic in Computer Science: Normative system specification.*, John Wiley and sons Ltd, 1993.
  37. David Makinson and Leendert W. N. van der Torre, ‘Input/output logics’, *Journal of Philosophical Logic*, 29(4), 383–408, (2000).
  38. D. Davidson, ‘The logical form of action sentences’, in *The Logic of Decision and Action*, ed., N. Rescher, Univ. of Pittsburgh Press, (1967).
  39. E. Bach, ‘On time, tense, and aspect: An essay in english metaphysics’, in *Radical Pragmatics*, ed., P. Cole, Academic Press, New York, (1981).
  40. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. (eds.) van der Torre,

Handbook of Deontic Logic and Normative Systems, College Publications, 2013.

41. X. Parent, 'Moral particularism in the light of deontic logic', *Artificial Intelligence and Law*, 19(2-3), 75–98, (2011). [25] L. Robaldo, 'Interpretation and inference with maximal referential terms.', *The Journal of Computer and System Sciences*, 76(5), (2010).
42. G. Sartor and E. Pattaro, *Legal Reasoning: A Cognitive Approach to the Law*, Treatise of legal philosophy and general jurisprudence, 2005.
43. M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory, 'The british nationality act as a logic program', *Communications of the ACM*, 29(5), (1986).
44. G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco, 'Computing strong and weak permissions in defeasible logic', *Journal of Philosophical Logic*, 6(42), (2013).
45. *Logic in the theory and practice of lawmaking*, eds., M. Araszkievicz and K. Pleszka, Springer, 2015.
46. Andrew J.I. Jones and Steven Orla Kimbrough, *A Note on Modelling Speech Acts as Signalling Conventions*, 325–342, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
47. A. Farrell, M. Sergot, M. Salle, and C. Bartolini, 'Using the event calculus for tracking the normative state of contracts', *International Journal of Cooperative Information Systems*, 14, 99–129, (2005).
48. M. Hashmi, G. Governatori, and M. Wynn, 'Modeling obligations with event-calculus', in *Rules on the Web. From Theory to Applications*, eds., A. Bikakis, P. Fodor, and D. Roman, volume 8620 of *Lecture Notes in Computer Science*, 296–310, Springer International Publishing, (2014).
49. R Kowalski and M Sergot, 'A logic-based calculus of events', *New Generation Computing*, 4(1), 67–95, (1986).
50. Antony Galton, 'Operators vs. arguments: The ins and outs of reification', *Synthese*, 150(3), 415–441, (2006).
51. Samit Khosla and T. S. E. Maibaum, 'The prescription and description of state based systems', in *Temporal Logic in Specification*, Altrincham, UK, April 8-10, 1987, *Proceedings*, eds., Behnam Banieqbal, Howard Barringer, and Amir Pnueli, volume 398 of *Lecture Notes in Computer Science*, pp. 243–294. Springer, (1987).
52. L. T. McCarty, 'Ownership: A case study in the representation of legal concepts', *Artificial Intelligence and Law*, 10(1-3), 135–161, (2002).
53. L. T. McCarty, 'Deep semantic interpretations of legal texts', in *The Eleventh*

International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA, pp. 217–224, (2007).

54. H. Kamp and U. Reyle, *From Discourse to Logic: an introduction to model-theoretic semantics, formal logic and Discourse Representation Theory*, Kluwer Academic Publishers, Dordrecht, 1993.
55. A. Copestake, D. Flickinger, and I.A. Sag, ‘Minimal Recursion Semantics. An introduction’, *Journal of Research on Language and Computation.*, 2(3), (2005).
56. J. R. Hobbs, ‘Deep lexical semantics’, in *Proc. of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, Haifa, Israel, (2008).
57. L. Robaldo, ‘Conservativity: a necessary property for the maximization of witness sets.’, *The Logic Journal of the IGPL*, 21(5), 853–878, (2013).
58. J.R. Hobbs, ‘The logical notation: Ontological promiscuity’, in *Chap. 2 of Discourse and Inference*, (1998). Available at <http://www.isi.edu/~hobbs/disinf-tc.html>.
59. L. Robaldo and E. Miltsakaki, ‘Corpus-driven semantics of concession: Where do expectations come from?’, *Dialogue&Discourse*, 5(1), (2014).
60. Doris Liebwald, ‘Vagueness in law: A stimulus for ’artificial intelligence & law’’, in *Proc. of the 14th International Conference on Artificial Intelligence and Law, ICAIL ’13*, pp. 207–211, (2013).
61. N. MacCormick and R.S. Summers, *Interpreting Statutes: A Comparative Study*, Applied legal philosophy, Dartmouth, 1991.
62. G. Governatori, A. Rotolo, and G. Sartor, ‘Deontic defeasible reasoning in legal interpretation’, in *The 15th International Conference on Artificial Intelligence & Law*, ed., K. Atkinson, San Diego, (2015).
63. J. McCarthy, ‘Circumscription: A form of nonmonotonic reasoning’, *Artificial Intelligence*, (13), 27–39, (1980).
64. Jorgen Jørgensen, ‘Imperatives and logic’, *Erkenntnis*, 7, (1937).
65. G. Governatori and M. Hashmi, ‘Permissions in deontic eventcalculus’, in *International Conference on Legal Knowledge and Information Systems (Jurix)*, pp. 181–182, Braga, Portugal, (2015).
66. Jorg Hansen, ‘Prioritized conditional imperatives: problems and a new ” proposal’, *Autonomous Agents and Multi-Agent Systems*, 17(1), (2008).
67. John Horty, *Reasons as Defaults*, Oxford University Press, 2012.

68. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, page 3, 2017
69. Закон України Про охорону праці
70. Закон України Про охорону навколишнього природного середовища
71. Закон України Про забезпечення санітарного та епідемічного благополуччя населення
72. Закон України Про відходи
73. Закон України Про охорону атмосферного повітря
74. Закон України Закон України „Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру”
75. Кодекс Водний кодекс України
76. Кодекс Кодекс законів про працю України
77. ДК 005-96 Державний класифікатор України. Класифікатор відходів
78. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно- обчислювальних машин
79. НПАОП 0.00-3.07-09 Норми безплатної видачі спеціального одягу, спеціального взуття та інших засобів індивідуального захисту працівникам загальних професій різних галузей промисловості
80. НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці
81. НПАОП 0.00-4.15-98 Про розробку інструкцій з охорони праці
82. НПАОП 0.00-6.03-93 Порядок опрацювання та затвердження власником нормативних актів про охорону праці
83. НПАОП 0.00-8.24-05 Перелік робіт з підвищеною небезпекою
84. НАПБ А. 01.001-2004 Правила пожежної безпеки України
85. НАПБ 01.022-89 Правила пожарной безопасности для объединений, предприятий и организаций Государственного комитета по вопросам вычислительной техники и информации СССР
86. НПАОП Б.02.005-2003 Про інструктаж, спецнавчання з питань пожежної безпеки

87. НАПБ Б.03.001-2004 Типових норм належності вогнегасників
88. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
89. НАПБ.Б.06.004-2005 Правила пожарной безопасности в Украине
90. НАПБ Б.07.005-86 Определение категорий помещений и зданий по взрывопожарной и пожарной опасности
91. НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок
92. НПАОП 40.1-1.21-98 Правила безпечної експлуатації електроустановок споживачів
93. НПАОП 40.1-1.32-01 Правила устройства электроустановок. Электрооборудование специальных установок
94. НПАОП 45.2-1.01-98 Правила обстеження технічного стану, оцінки технічного стану та паспортизації виробничих будівель і споруд
95. НПАОП 45.2-4.01-98 Положення про безпечну і надійну експлуатацію виробничих будівель і споруд
96. НПАОП 72.0-1.01-75 Правила по охране труда, технике безопасности и производственной санитарии для вычислительных центров, машиносчетных станций, районных ( городских ) информационно-вычислительных станций ( центров ) ЦСУ СССР рус
97. ДБН А.1.2.12-2008 Система надійності та безпеки в будівництві
98. ДБН В.1.1.7-2002 Пожежна безпека об'єктів будівництва
99. ДБН В.1.2.7-2007 Основні вимоги до будівель та споруд
100. ДБН В.2.5-13-98 Пожежна автоматика будинків і споруд.
101. ДБН В.2.5-28:2015 Природне і штучне освітлення
102. ДБН В.2.5-56:2014 Системи протипожежного захисту
103. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування
104. ДБН А.3.2.2-2009 Система стандартів безпеки праці. Охорона праці і промислова безпека у будівництві. Основні положення
105. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку
106. ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної

- вібрації
107. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих
108. ДСанПіН 2.2.7.029 Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення
109. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин
110. СанПіН 4947-89 Временные санитарно-гигиенические требования к кооперативам и индивидуальной трудовой деятельности по производству, ремонту и наладке промышленных изделий и бытовой техники, а также в сфере кустарно-ремесленных промыслов
111. СанПіН 5804-91 Санитарные нормы и правила устройства и эксплуатации лазеров
112. СНиП II-4-79 Естественное и искусственное освещение
113. ДСТУ ІЕС/ISO 31010:2013 Керування ризиком. Методи загального оцінювання ризику (ІЕС/ISO 31010:2009, IDT)
114. ДСТУ ISO 14001:2006 Системы экологического управления. Требования и руководящие указания по применению (ISO 14001:2004)
115. ISO 14050:2002 Системы экологического управления. Термины и определения. Глоссарий
116. ДСТУ 3911-99 (ГОСТ 17.9.0.1-99) Охорона природи. Поводження з відходами. Виявлення відходів і подання інформаційних даних про відходи. Загальні вимоги
117. ДСТУ-Н Б Б.2.2-7:2013 Настанова з улаштування контейнерних майданчиків
118. ДСТУ 4462.0.02:2005 Охорона природи. Комплекс стандартів у сфері поводження з відходами. Загальні вимоги
119. ДСТУ 4462.3.02:2006 Охорона природи. Поводження з відходами. Пакування, маркування і захоронення відходів. Правила перевезення відходів. Загальні технічні та організаційні вимоги
120. ДСТУ 4462.3.01:2006 Охорона природи. Поводження з відходами. Порядок здійснення операцій

121. ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация
122. ГОСТ 2.1.004-91 ССБТ. Пожарная безопасность. Общие требования
123. ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны
124. ГОСТ 12.1.001-89 ССБТ. Ультразвук. Общие требования безопасности
125. ГОСТ 12.1.003-83 ССБТ. Шум. Общие требования безопасности
126. ГОСТ 12.1.004-91. ССБТ. Пожарная безопасность. Общие требования
127. ГОСТ 12.1.005-88. ССБТ. Загальні санітарно-гігієнічні вимоги до повітряу робочої зони
128. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности. Допустимые уровни на рабочих местах и требования к проведению контроля.
129. ГОСТ 12.1.007-76 ССБТ. Вредные вещества. Классификация и общие требования безопасности.
130. ГОСТ 12.1.010-76 ССБТ. Взрывобезопасность. Общие требования.
131. ГОСТ 12.1.011-78 ССБТ. Смеси взрывоопасные. Классификация и методы испытаний.
132. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вибрационная безопасность. Общие требования
133. ГОСТ 12.1.018-93 ССБТ. Пожаровзрывобезопасность статического электричества. Общие требования
134. ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление
135. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
136. ГОСТ 12.2.007.0-75 ССБТ. Изделия электротехнические. Общие требования безопасности.
137. ГОСТ 12.2.032-78 Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования
138. ГОСТ 12.4.009-83. ССБТ. Пожарная техника для защиты объектов. Основные виды. Размещение и обслуживание
139. ГОСТ 13109-97 „Электрическая энергия. Совместимость технических средств электромагнитных. Нормы качества



- электроэнергоснабжения общего назначения”
140. ГОСТ 25861-83 Машины вычислительные и системы обработки данных. Требования по электрической и механической безопасности и методы испытаний
141. N 387/22-78 від 03.10.1986 Типовое положение об оценке условий труда на рабочих местах и порядок применения отраслевых перечней работ, на которых могут устанавливаться доплаты рабочим за условия труда: постановление Госкомтруда СССР и Секретариата ВЦСПС
142. Безопасные уровни содержания вредных веществ в окружающей среде. ГосНИИБХП, г.Северодонецк, 1994 г.
143. Ткачук К.Н., Иванчук Д.Ф. и др. Справочник по охране труда на промышленном предприятии. –К.: «Техника», 1991 г.
144. Долин П.А. Справочник по технике безопасности. –М.:»Энергоатомиздат», 1984 г.
145. Вредные химические вещества. Неорганические соединения элементов/ I-IV групп: Справ. изд. /Под ред. В.А.Филова и др.-Л: Химия, 1986.-512с.
146. Вредные химические вещества. Неорганические соединения элементов/ V-VIII групп: Справ. изд. /Под ред. В.А.Филова и др.-Л: Химия, 1989.-592с.
147. Свойства неорганических соединений элементов. Справочник/ Ефимов А.И. и др.-Л: Химия, 1983.-392с.
148. Беспямятов Г.П., Кротов Ю.А. Предельно-допустимые концентрации химических веществ в окружающей среде. Справочник.-Л.:Химия,1985.- 528с.
149. Вредные вещества в промышленности. Справочник. Изд. 7-е пер. и доп. в 3-х томах. Том 3. Неорганические вещества /Под ред. Н.В.Лазарева. - Л.:Химия,1976.- 608 с.
150. Пожаровзрывоопасность веществ и материалов и средства их тушения. Справочник, Под ред. Баратова А.Н., в 2-х томах, М.: Химия, 1990.
151. Добровольский А.А., Переслыцких Ф.Ф.. Пожарная техника. Справочник. К.: Техника, 1981.
152. Правила устройства электроустановок (ПУЭ). М.: Энергоатомиздат, 1985 г.
153. Кобевник В.Ф. Охрана труда. Учебное пособие для вузов. – К.: «Высшая школа», 1990 г.
154. Безопасность жизнедеятельности. Под ред. С.В.Белова, М.:Высшая школа, 1999 г.

155. Охрана труда в химической промышленности. Учебник для вузов. Под ред. Макарова Г.В. – М.:»Химия», 1989 г.
156. Справочная книга по светотехнике / Под ред. Ю. Б. Айзенберга. 3-е изд. перераб. и доп. — М.: Знак. 2006.— 972 с.
157. Справочная книга для проектирования электрического освещения / Под ред. Г.М. Кнорринга.- М.: Энергия, 1976. - 384 с.

## Додаток А

## Лістинг програмного коду семантико-синтаксичного аналізатора

## Лістинг result\_data.py:

```
# -*- coding: utf-8 -*-

import collections
import io
import urllib
import re
import os
import requests

from odoo import api, fields, models, _
from odoo.exceptions import UserError, ValidationError
from odoo.tools import float_round
from conllu.parser import parse, parse_tree
from json import dumps, loads
from .app import dragnn_parser

from flask import Flask
from flask_cors import CORS
from flask_restful import Resource, Api, reqparse

parse_handler = None

app = Flask(__name__)
CORS(app)
```

```

class MoAnalysticTextSentence(models.Model):

    _name = 'mo.analystic.text.sentence'

    _description = 'Results analysis of the text (sentence)'

    text_result_id = fields.Many2one('mo.analystic.text.result', string='Analysis of the text',
readonly=True, required=True)

@api.one
def _compute_result_diagram(self):

def instance_res(result_data, result, id_d, level, parent_id):

    my_level = level + 1

    my_parent_id = parent_id

    for data in result_data:

        if isinstance(data, dict):

            result["nodeDataArray"].append({

                "id": id_d,

                "loc": str(my_level * 60) + " " + str(id_d * 60),

                "text": data["form"],

            })

        if id_d != parent_id:

            result["linkDataArray"].append({

                "from": id_d,

                "to": parent_id,

```

```

        "text": "",
        "curviness": 20,

    })

    my_parent_id = id_d
    id_d += 1

else:
    id_d = instance_res(data, result, id_d, my_level, my_parent_id)

return id_d

```

for analys in self:

```

result = {
    "nodeKeyProperty": "id",
    "nodeDataArray": [],
    "linkDataArray": []
}

if self.result_serial:
    result_data = loads(self.result_serial)

    id_d = 0
    level = 0
    parent_id = 0

    id_d = instance_res(result_data, result, id_d, level, parent_id)

result_diagram = dumps(result, ensure_ascii=False)

analys.result_diagram = result_diagram

```

```
name = fields.Char('Name')
```

```

source_text = fields.Text('Source Text')
result_text = fields.Text('Result Text')
result_serial = fields.Text('Result value')
result_diagram = fields.Text(compute=_compute_result_diagram)

```

```
@api.multi
```

```

def name_get(self):
    if self.source_text or self.source_text == "":
        return self.source_text
    else:
        return '<None>'

```

```
class MoAnalysticTextResult(models.Model):
```

```

    _name = 'mo.analystic.text.result'
    _description = 'Results analysis of the text'
    _order = "parent_left, sequence, name"
    _parent_order = "sequence, name"
    _parent_store = True

```

```
sequence = fields.Integer('Sequence')
```

```
parent_id = fields.Many2one('mo.analystic.text.result', string='Parent Table Of Content',
ondelete='cascade')
```

```
child_ids = fields.One2many('mo.analystic.text.result', 'parent_id', string='Children Table Of Content')
```

```
parent_left = fields.Integer(string='Left Parent', index=True)
```

```
parent_right = fields.Integer(string='Right Parent', index=True)
```

```
@api.model_cr
```

```
def init(self):
```

```
    path_model = os.path.realpath(__file__).replace('result_data.py','models')
```

```
    parse_handler = dragnn_parser.SyntaxnetParser("{} / {}".format(path_model, 'English'))
```

```
@api.one
```

```
def _compute_result_diagram(self):
```

```
    #
```

```
    #     result_diagram = ""
```

```
    # { "nodeKeyProperty": "id",
```

```
    #   "nodeDataArray": [
```

```
    #     { "id": 0, "loc": "120 120", "text": "Initial" },
```

```
    #     { "id": 1, "loc": "330 120", "text": "First down" },
```

```
    #     { "id": 2, "loc": "226 376", "text": "First up" },
```

```
    #     { "id": 3, "loc": "60 276", "text": "Second down" },
```

```
    #     { "id": 4, "loc": "226 226", "text": "Wait" }
```

```
    # ],
```

```
    # "linkDataArray": [
```

```
    #   { "from": 0, "to": 0, "text": "up or timer", "curviness": -20 },
```

```
    #   { "from": 0, "to": 1, "text": "down", "curviness": 20 },
```

```
    #   { "from": 1, "to": 0, "text": "up (moved)\nPOST", "curviness": 20 },
```

```
    #   { "from": 1, "to": 1, "text": "down", "curviness": -20 },
```

```
    #   { "from": 1, "to": 2, "text": "up (no move)" },
```

```
    #   { "from": 1, "to": 4, "text": "timer" },
```

```
    #   { "from": 2, "to": 0, "text": "timer\nPOST" },
```

```
    #   { "from": 2, "to": 3, "text": "down" },
```

```
    #   { "from": 3, "to": 0, "text": "up\nPOST\n(dblclick\nif no move)" },
```

```
# { "from": 3, "to": 3, "text": "down or timer", "curviness": 20 },
# { "from": 4, "to": 0, "text": "up\nPOST" },
# { "from": 4, "to": 4, "text": "down" }
# ]
# }"
```

```
result = {
    "nodeKeyProperty": "id",
    "nodeDataArray": [],
    "linkDataArray": []
}
```

```
def instance_res(analys):

    if analys.result_serial:
        result_data = loads(analys.result_serial)
    else:
        return

    token_str = list()
    children = [[] for token in result_data['output']]
    root = -1

    for data in result_data['output']:
        data ['contains'] = []
        token_str.append(data)
        if data['head'] == -1:
            root = data['id']
```



```

else:
    children[data['head']].append(data['id'])

def _get_dict(parent):
    d = []
    for c in children[parent['id']]:
        dict_c = dict(token_str[c])
        parent['contains'].append(dict_c)
        _get_dict(dict_c)

tree = []
tree.append(dict(token_str[root]))
_get_dict(tree[0])

def append_node_to(data_tree, id_d, level, parent_id):
    my_level = level + 1
    my_parent_id = parent_id
    for data in data_tree:
        dep = data['dep']
        word = data["word"]
        if dep == 'obj':
            word = word + u'\n(об\ект\пправовідносин)'
        elif dep == 'nsubj':
            word = word + u'\n(суб\ект\пправовідносин)'

    result["nodeDataArray"].append({
        "id": id_d,
        "loc": str(data['id'] * 80) + " " + str(my_level * 60),

```

```

        "text": word,
    })
    if id_d != parent_id:
        from_id = id_d
        to_id = parent_id
        if data['id'] > parent_id:
            from_id = parent_id
            to_id = id_d
        result["linkDataArray"].append({
            "from": from_id,
            "to": to_id,
            "text": data['dep'],
            "curviness": 20,

        })
    my_parent_id = id_d
    id_d += 1
    contains = data.get('contains')
    if contains and len(contains) > 0:
        id_d = append_node_to(contains, id_d, my_level, my_parent_id)
    return id_d

if analys.result_serial:
    id_d = 0
    level = 0
    parent_id = 0
    id_d = append_node_to(tree, id_d, level, parent_id)

```

```

for analys in self:
    instance_res(analys)
    result_diagram = dumps(result, ensure_ascii=False)
    analys.result_diagram = result_diagram

name = fields.Char('Name', required=True, translate=True)
source_text = fields.Text('Source Text')
result_text = fields.Text('Result Text')
result_serial = fields.Text('Result value')
result_diagram = fields.Text(compute=_compute_result_diagram, default = '{"nodeKeyProperty":
"id","nodeDataArray": [],"linkDataArray": []}')

@api.constrains('parent_id')
def check_parent_id(self):
    if not self._check_recursion():
        raise ValueError(_('Error ! You cannot create recursive categories.))

@api.multi
def name_get(self):
    res = []
    for record in self:
        name = record.name
        if record.parent_id:
            name = record.parent_id.name + '/' + name
        res.append((record.id, name))

```

```
return res
```

```
@api.one
```

```
def client_action_parse(self):
```

```
    normalize_data = re.sub(r'\s+', ' ', self.source_text)
```

```
    normalize_data = re.sub(r'\s\.', '.', normalize_data)
```

```
    normalize_data = re.sub(r'(\.)', '!', normalize_data)
```

```
    normalize_data = re.sub(r'\s\+', '!', normalize_data)
```

```
    normalize_data = re.sub(r'(\,)', '!', normalize_data)
```

```
    normalize_data = re.sub(r'\s\;+', '!', normalize_data)
```

```
    normalize_data = re.sub(r'(\;)', '!', normalize_data)
```

```
    normalize_data = re.sub(r'\s\!+', '!', normalize_data)
```

```
    normalize_data = re.sub(r'(\!)', '!', normalize_data)
```

```
    normalize_data = re.sub(r'\s\?+', '?', normalize_data)
```

```
    normalize_data = re.sub(r'(\?)', '?', normalize_data)
```

```
    data_req = {
```

```
        "strings": [normalize_data],
```

```
        "tree": False
```

```
    }
```

```
    global parse_handler
```

```

if parse_handler == None:
    path_model = os.path.realpath(__file__).replace('result_data.py', 'models')
    #parse_handler = dragnn_parser.SyntaxnetParser("{} / {}".format(path_model, 'Ukrainian'))
    parse_handler = dragnn_parser.SyntaxnetParser("{} / {}".format(path_model, 'Russian-
SynTagRus'))
    #parse_handler = dragnn_parser.SyntaxnetParser("{} / {}".format(path_model, 'English'))

data_trans = [normalize_data]
res = parse_handler.parse_multi_string(data_trans)
self.result_text = res[0]['text_colln']
self.result_serial = dumps(res[0]['result'], ensure_ascii=False)
pass

```

#### **Листинг dragnn\_parser.py:**

```

#!/usr/bin/env python

import copy
import os
import pprint
import subprocess
import sys

import re

from config import config

import collections

```

```

import os

import tensorflow as tf

from dragnn.protos import spec_pb2

from dragnn.python import graph_builder

from dragnn.python import spec_builder

from dragnn.python import load_dragnn_cc_impl # This loads the actual op definitions

from dragnn.python import render_parse_tree_graphviz

from dragnn.python import visualization

from google.protobuf import text_format

from syntaxnet import load_parser_ops # This loads the actual op definitions

from syntaxnet import sentence_pb2

from syntaxnet.ops import gen_parser_ops

from tensorflow.python.platform import tf_logging as logging

import asciitree

class SyntaxnetParser(object):

    def __init__(self,model_folder, folder=config.syntaxnetFolder):

        self.folder=folder

        reload(sys)

        sys.setdefaultencoding('utf-8')

        self.segmenter_model = self.load_model("{} /segmenter".format(model_folder), "spec.textproto",
"checkpoint")

        self.parser_model = self.load_model(model_folder, "parser_spec.textproto", "checkpoint")

    def load_model(self,base_dir, master_spec_name, checkpoint_name):

        # Read the master spec

        master_spec = spec_pb2.MasterSpec()

```

```

with open(os.path.join(base_dir, master_spec_name), "r") as f:
    text_format.Merge(f.read(), master_spec)
spec_builder.complete_master_spec(master_spec, None, base_dir)
logging.set_verbosity(logging.WARN) # Turn off TensorFlow spam.

# Initialize a graph
graph = tf.Graph()
with graph.as_default():
    hyperparam_config = spec_pb2.GridPoint()
    builder = graph_builder.MasterBuilder(master_spec, hyperparam_config)
    # This is the component that will annotate test sentences.
    annotator = builder.add_annotation(enable_tracing=True)
    builder.add_saver() # "Savers" can save and load models; here, we're only going to load.

sess = tf.Session(graph=graph)
with graph.as_default():
    # sess.run(tf.global_variables_initializer())
    # sess.run('save/restore_all', {'save/Const:0': os.path.join(base_dir, checkpoint_name)})
    builder.saver.restore(sess, os.path.join(base_dir, checkpoint_name))

def annotate_sentence(sentence):
    with graph.as_default():
        return sess.run([annotator['annotations'], annotator['traces']],
                        feed_dict={annotator['input_batch']: [sentence]})

    return annotate_sentence

def to_colln(self, sentence):

```

"""Builds a dictionary representing the parse tree of a sentence.

Note that the suffix "@id" (where 'id' is a number) is appended to each element to handle the sentence that has multiple elements with identical representation. Those suffix needs to be removed after the asciitree is rendered.

Args:

sentence: Sentence protocol buffer to represent.

Returns:

Dictionary mapping tokens to children.

"""

```
def parse_attribute(input, todo_dict):
```

```
    pattern = re.compile(r\"^[^]*\")
```

```
    result = list(pattern.findall(input))
```

```
    #print(result)
```

```
    return_val = {}
```

```
    for i in range(0, len(result), 2):
```

```
        print(filter(None, result[i].split(\"\")))
```

```
        todo_dict[filter(None, result[i].split(\"\"))[0]] = filter(None, result[i + 1].split(\"\"))[0]
```

```
    return todo_dict
```

```
token_str = list()
```

```
token_str.append('#ID\tFORM\tLEMMA\tUPODTAG\tXPOSTAG\tFEATS\tHEAD\tDEPREL\tDEPS\tMISC')
```

```
#children = [[] for token in sentence.token]
```

```
root = -1
```

```
for i in range(0, len(sentence.token)):
```

```
    token = sentence.token[i]
```



```

tags = {}

tags = parse_attribute(str(token.tag), tags)

arr_feats = []

for tag in tags.keys():
    if tag == 'fPOS':
        continue

    arr_feats.append(tag + '=' + tags[tag])

str_feats = '|'.join(arr_feats)

token_str.append('%d\t%s\t%s\t%s\t%s\t%s\t%d\t%s\t%s\t%s' % ((i + 1),
token.word, '_', tags['fPOS'], '_', str_feats, token.head + 1, token.label, '_', '_'))

# if token.head == -1:

#   root = i

# else:

#   children[token.head].append(i)

# def _get_dict(i):

#   d = collections.OrderedDict()

#   for c in children[i]:

#       d[token_str[c]] = _get_dict(c)

#   return d

return token_str

def annotate_text(self, text):

    sentence = sentence_pb2.Sentence(

        text=text,

        token=[sentence_pb2.Token(word=text, start=-1, end=-1)]

    )

```

```

# preprocess
with tf.Session(graph=tf.Graph()) as tmp_session:
    char_input = gen_parser_ops.char_token_generator([sentence.SerializeToString()])
    preprocessed = tmp_session.run(char_input)[0]
    segmented, _ = self.segmenter_model(preprocessed)

    annotations, traces = self.parser_model(segmented[0])
    assert len(annotations) == 1
    assert len(traces) == 1
    return sentence_pb2.Sentence.FromString(annotations[0])

def exec_from_syntax(self,string):
    os.chdir(self.folder)
    my_env = os.environ
    my_env['PYTHONIOENCODING'] = 'utf-8'
    p = subprocess.Popen([
        "syntaxnet/demo.sh"
    ], shell=True, stderr=subprocess.PIPE, stdin=subprocess.PIPE, stdout=subprocess.PIPE,
    env=my_env)
    output, err = p.communicate(string)
    return output
"""
1   Google _   ADJ   JJ   Number=Sing|fPOS=PROPN++NNP   3   nsubj _   _
2   is _   VERB   VBZ
Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBZ   3   cop _
_
3   awesome! _   NOUN   NN   Degree=Pos|fPOS=ADJ++JJ   0   ROOT _   _
"""

```

```

def parse_notree_string(self,output_list):
    """Used for parsing the output of a none tree stuff.."""
    # lines=len(output_list)

    output = []

    for i in range(0, len(output_list['output'])):
        output.append({})
        i = 0

    for stuff in output_list['output']:
        print(stuff)
        output[i] = stuff
        i = i + 1

    # Start to merge it to a tree
    structured_tree = {}
    print(output)

def merge_to_target(target, output, tree):
    for i in range(0, len(output_list['output'])):
        if 'head' in output[i] and output[i]['head'] == target:
            del output[i]['head']
            if 'contains' in tree:
                tree['contains'].append(output[i])
            else:
                tree['contains'] = [output[i]]
            merge_to_target(i, output, tree['contains'][tree['contains'].index(output[i])])

    for i in range(0, len(output_list['output'])):

```

```

if 'head' in output[i] and output[i]['head'] == -1:
    structured_tree = output[i]
    del output[i]['head']
    merge_to_target(i, output, structured_tree)
return structured_tree

```

```
def parse_string_from_draggn(self,sentence):
```

```
    result = self.annotate_text(sentence)
```

```
def parse_attribute(input, todo_dict):
```

```
    pattern = re.compile(r'^"[^"]*\\"')

```

```
    result = list(pattern.findall(input))

```

```
    #print(result)

```

```
    return_val = {}

```

```
    for i in range(0, len(result), 2):

```

```
        print(filter(None, result[i].split("")))

```

```
        todo_dict[filter(None, result[i].split("))[0].lower()] = filter(None, result[i + 1].split("))[0]

```

```
    return todo_dict

```

```
array_colln = self.to_colln(result)
```

```
print 'Input: %s' % result.text
```

```
print 'Parse:'
```

```
for tr_ln in array_colln:
```

```
    print tr_ln

```

```
text_colln = '\n'.join(array_colln)
```

```
return_dict = {}

```

```
return_dict['input'] = result.text

```

```
output = []

```

```

index = -1

for stuff in result.token:

    index += 1

    tmpdict = {}

    tmpdict['id'] = index

    tmpdict['word'] = stuff.word

    tmpdict['label'] = stuff.label

    tmpdict['dep'] = stuff.label

    tmpdict['break_level'] = stuff.break_level

    tmpdict['category'] = stuff.category

    tmpdict['head'] = stuff.head

    tmpdict = parse_attribute(str(stuff.tag), tmpdict)

    # tmp fix for getting back the pos_tag

    tmpdict['pos_tag']=tmpdict['fpos'].split('++')

    output.append(tmpdict)

return_dict['output'] = output

return return_dict, text_colln

```

```

def parse_multi_string(self,string_list):

```

```

    def generate():

        string=""

        for stuff in string_list:

            if string:

                string=string+'\n'+stuff

            else:

                string=stuff

```

```

return string

try:
    return_list=[]
    for string in string_list:
        print("Annotating: " + string)
        result, text_colln = self.parse_string_from_dragnn(string)
        return_list.append({
            'result': result,
            'text_colln': text_colln
        })
    result_json=return_list
except Exception as e:
    print(e)
    result_json={"status":"error","reason":e}
finally:
    return result_json

def parse_string(self,format):
    def parse_col(data, json):
        if data[0] == '|':
            if 'contains' not in json[len(json) - 1]:
                json[len(json) - 1]['contains'] = []
            json[len(json) - 1]['contains'] = parse_col(data[3:], json[len(json) - 1]['contains'])
        elif data[0] == '+--':
            builder = {"name": data, "pos_tag": data[15], "dep": data[3-]}
            json.append(copy.copy(builder))
        elif (data[0] == ") and (data==" and (data[15]=="") and (data[3-]==""):

```

```

    if 'contains' not in json[len(json) - 1]:
        json[len(json) - 1]['contains'] = []
    json[len(json) - 1]['contains'] = parse_col(data[4:], json[len(json) - 1]['contains'])

    return json

json = {}

#format=['Input: The quick brown fox jumps over the lazy dog', 'Parse:', 'jumps VBZ ROOT', ' +--
fox NN nsubj', '| +-- The DT det', '| +-- quick JJ amod', '| +-- brown JJ amod', ' +-- over IN prep', '
+-- dog NN pobj', ' +-- the DT det', ' +-- lazy JJ amod', "]

    key = format[15].split(' ')

    json = {'name': key[0], "pos_tag": key, "dep": key[15], "contains": []}

    for lines in format[3:]:
        if lines=="":
            pass
        else:
            data = lines.split(' ')[1:]
            json['contains']=parse_col(data, json['contains'])

    pprint.pprint(json)

    return json

```

### Листинг mo\_analytic\_result\_data\_view.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<odoo>
  <data>
    <record id="view_mo_analytic_text_result_tree" model="ir.ui.view">
      <field name="name">mo.analytic.text.result.tree</field>
      <field name="model">mo.analytic.text.result</field>
      <field name="priority" eval="8" />
      <field name="field_parent" eval="False"/>
      <field name="arch" type="xml">

```

```

    <tree string="Analysis Results">
      <field name="sequence" widget="handle"/>
      <field name="display_name"/>
    </tree>
  </field>
</record>
<record id="view_mo_analystic_text_result_form" model="ir.ui.view">
  <field name="name">mo.analystic.text.result.form</field>
  <field name="model">mo.analystic.text.result</field>
  <field name="priority" eval="8" />
  <field name="arch" type="xml">
    <form string="Analysis Result" onload="init()">
      <sheet>
        <button name="client_action_parse" icon="fa-refresh" string="Generate Result"
type="object" class="oe_highlight" />
        <group col="2">
          <field name="parent_id" />
          <field name="name" />
          <field name="sequence" />
          <field name="source_text" />
          <!-- <field name="result_diagram" /> -->
        </group>
      </sheet>
    </form>
  </field>
</record>
<div id="myDiagramDiv" style="border: solid 1px black; width: 100%; height: 400px"></div>
  <field name="result_text" />
  <field name="result_diagram" />
  <!--
{ "nodeKeyProperty": "id",
  "nodeDataArray": [

```



```

    { "id": 0, "loc": "120 120", "text": "Initial" },
    { "id": 1, "loc": "330 120", "text": "First down" },
    { "id": 2, "loc": "226 376", "text": "First up" },
    { "id": 3, "loc": "60 276", "text": "Second down" },
    { "id": 4, "loc": "226 226", "text": "Wait" }
  ],
  "linkdataArray": [
    { "from": 0, "to": 0, "text": "up or timer", "curviness": -20 },
    { "from": 0, "to": 1, "text": "down", "curviness": 20 },
    { "from": 1, "to": 0, "text": "up (moved)\nPOST", "curviness": 20 },
    { "from": 1, "to": 1, "text": "down", "curviness": -20 },
    { "from": 1, "to": 2, "text": "up (no move)" },
    { "from": 1, "to": 4, "text": "timer" },
    { "from": 2, "to": 0, "text": "timer\nPOST" },
    { "from": 2, "to": 3, "text": "down" },
    { "from": 3, "to": 0, "text": "up\nPOST\n(dblclick\nif no move)" },
    { "from": 3, "to": 3, "text": "down or timer", "curviness": 20 },
    { "from": 4, "to": 0, "text": "up\nPOST" },
    { "from": 4, "to": 4, "text": "down" }
  ]
}

-->
    <script type="text/javascript">
        init();
    </script>
</sheet>
</form>
</field>

```

</record>

<record id="view\_mo\_analystic\_text\_result\_search" model="ir.ui.view">

<field name="name">mo.analystic.text.result.search</field>

<field name="model">mo.analystic.text.result</field>

<field name="priority" eval="8" />

<field name="arch" type="xml">

<search string="Analysis Results">

<field name="name" />

</search>

</field>

</record>

<record model="ir.actions.act\_window" id="action\_open\_mo\_analystic\_text\_result\_view">

<field name="name">Results Analysis</field>

<field name="type">ir.actions.act\_window</field>

<field name="res\_model">mo.analystic.text.result</field>

<field name="view\_type">form</field>

<field name="view\_mode">tree,form</field>

<field name="search\_view\_id" ref="view\_mo\_analystic\_text\_result\_search" />

<field name="domain">[]</field>

<field name="context">{}</field>

</record>

<record model="ir.actions.act\_window.view"  
id="action\_open\_mo\_analystic\_text\_result\_view\_form">

<field name="act\_window\_id" ref="action\_open\_mo\_analystic\_text\_result\_view" />

<field name="sequence" eval="20" />

<field name="view\_mode">form</field>

<field name="view\_id" ref="view\_mo\_analystic\_text\_result\_form" />

</record>

```
<record model="ir.actions.act_window.view"
id="action_open_mo_analytic_text_result_view_tree">
    <field name="act_window_id" ref="action_open_mo_analytic_text_result_view" />
    <field name="sequence" eval="10" />
    <field name="view_mode">tree</field>
    <field name="view_id" ref="view_mo_analytic_text_result_tree" />
</record>
</data>
</odoo>
```

Додаток Б  
Слайди комп'ютерної презентації



## Програмно-алгоритмічні засоби семантико-синтаксичного аналізу законодавчих документів

Магістрська атестійна робота  
Касьянов М.С.



## Актуальність

- Пошуку протиріч у законодавчих нормах
- Уникання матеріальних втрат у зв'язку з правовою неосвіченістю
- Протидія корупції в органах судової та державної влади

## Завдання проекту

- Вибрати й описати технологію для створення логічного аналізатора юридичних документів.
- Провести синтаксичний аналіз законодавчого тексту.
- Провести семантичний аналіз законодавчого тексту

## Формалізація прикладу логічного аналізатора

- Адвокат, який бажає практикувати в якій-небудь державі-члені, відмінному від того, в якому він отримав свою професійну кваліфікацію, реєструється в компетентному органі в цій державі.
- Коли пекарський мед використовується в якості інгредієнта в складному харчовому продукті, термін «мед» може використовуватися в назві продукту складової їжі замість терміна «пекарський мед».

## Принцип композиційності Монтегю

«Джон вважає, що Джек хоче їсти морозиво»

*вважати*[Джон,  
           *хотіти*(Джек,  
                    $\exists x[(\text{морозиво } x) \wedge (\text{їсти Джек } x)]$ ]  
 ]

## Предиката *Rexist* Хоббса

Для всіх  $e$  і  $e_1$  таких, що виконується ( $\text{not}' e e_1$ ), також має місце

$$(\text{Rexist } e) \leftrightarrow \neg (\text{Rexist } e_1)$$

## Імовірні або фактичні обставини Хоббса

«Джон вважає, що Джек хоче їсти морозиво»

$$\exists_e \exists_{e_1} \exists_{e_2} \exists_{ex} [(Rexist\ e) \wedge (вважати'\ e\ Джон\ e_1) \wedge (хотіти'\ e_1\ Джек\ e_2) \wedge (їсти'\ e_3\ Джек\ x) \wedge (морозиво'\ e_3\ x)]$$

## Всі юристи - це люди

Для всіх  $e_1, x$  таких, що мають місце (юрист'  $e_1\ x$ ), також має місце

$$\exists_{ei} \exists_{e_2} [(imply'\ e_i\ e_1\ e_2) \wedge (Rexist\ e_i) \wedge (люди'\ e_2\ x)]$$

## Формалізація прикладу обов'язку

*Адвокат, який бажає практикувати в якій-небудь державі-члені, відмінному від того, в якому він отримав свою професійну кваліфікацію, реєструється в компетентному органі в цій державі*

$$\forall x \forall y (\exists e_1 \exists e_2 [(Rexist\ e_1) \wedge (\text{адвокат } x) \wedge (\text{ДЧ } y) \wedge (\text{бажати}'\ e_1\ x\ e_2) \wedge (\text{практикувати}'\ e_2\ x) \wedge (\text{в } e_2\ y) \wedge \text{diffFrom}(y\ f_1(x))], \exists e_3 [(Rexist\ e_3) \wedge (\text{реєструватись}'\ e_3\ x) \wedge (\text{в } e_3\ f_2(y))])$$

## Формалізація прикладу дозволу

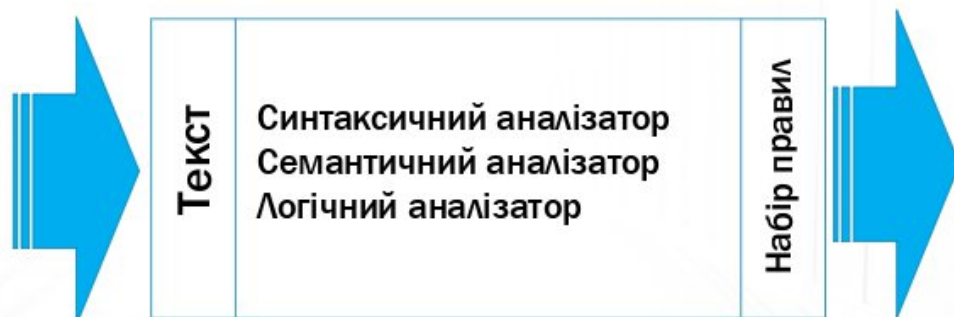
*Коли пекарський мед використовується в якості інгредієнта в складному харчовому продукті, термін «мед» може використовуватися в назві продукту складової їжі замість терміна «пекарський мед».*

$$\forall y (\exists x \exists e_1 [(Rexist\ e_1) \wedge (\text{інгредієнт}'\ e_1\ x\ y) \wedge (\text{пекарськийМед } x) \wedge (\text{харчовийПродукт } y)], \exists e_2 [(Rexist\ e_2) \wedge (\text{замінювати}'\ e_2\ T_h\ T_{bh}) \wedge (\text{в } e_2\ f_3(y))])$$

Зверніть увагу, що змінна  $x$  зустрічається тільки



## Глибокий семантико-синтаксичний аналізатор



## Інтерфейс користувача. Список.

Analysis of the text Results Analysis

Analysis of the text  
Додатки  
Налаштування

Results Analysis

СТВОРИТИ ІМПОРТ

- Display Name
- + Конституція України
- + Конституція України / Раздел 1 - Общие положения Конституции Украины
- + Раздел 1 - Общие положения Конституции Украины / Статья 13.
- + Раздел 1 - Общие положения Конституции Украины / Статья 13.
- + Раздел 1 - Общие положения Конституции Украины / Статья 14.

# Інтерфейс користувача. Форма.

Result Analysis / Різниця 1 - Обласні органи Кабінету Міністрів України / Сторінка 1

Батьківський елемент

Найменування документа (розділу)

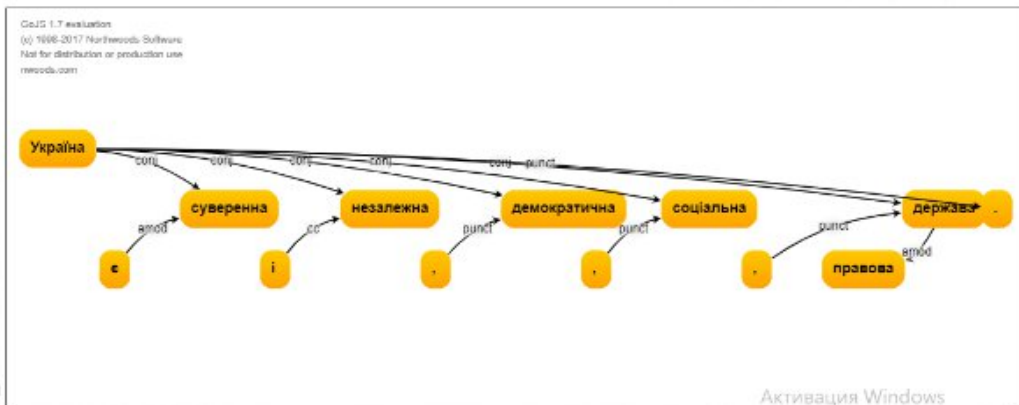
Вхідний текст

Результат аналізу у вигляді діаграми

Результат аналізу в форматі CoNLL-U

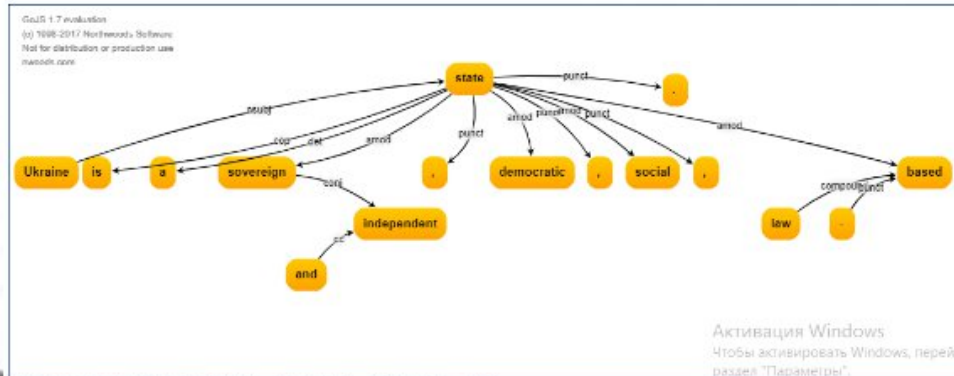
# Синтаксичний аналіз. Українська мовна модель

*"Україна є суверенна і незалежна, демократична, соціальна, правова держава."*



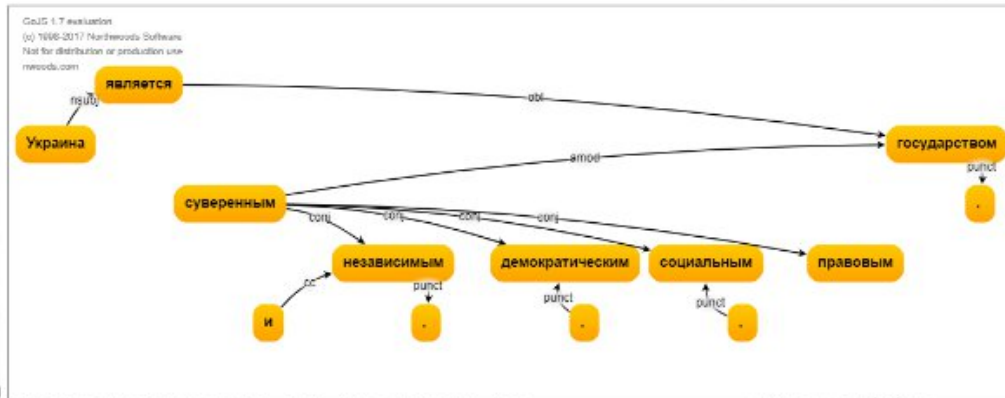
## Синтаксичний аналіз. Англійська мовна модель

*"Ukraine is a sovereign and independent, democratic, social, law-based state."*

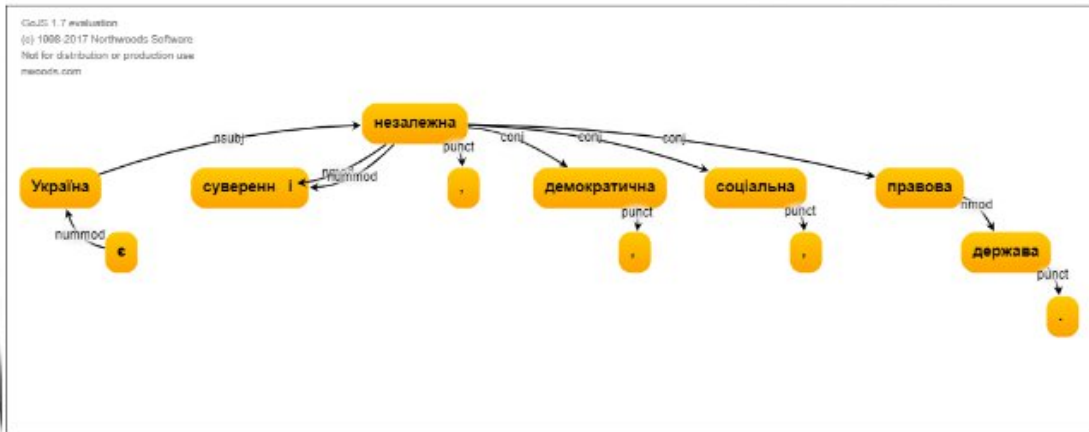


## Синтаксичний аналіз. Російська мовна модель

*"Украина является суверенным и независимым, демократическим, социальным, правовым государством."*

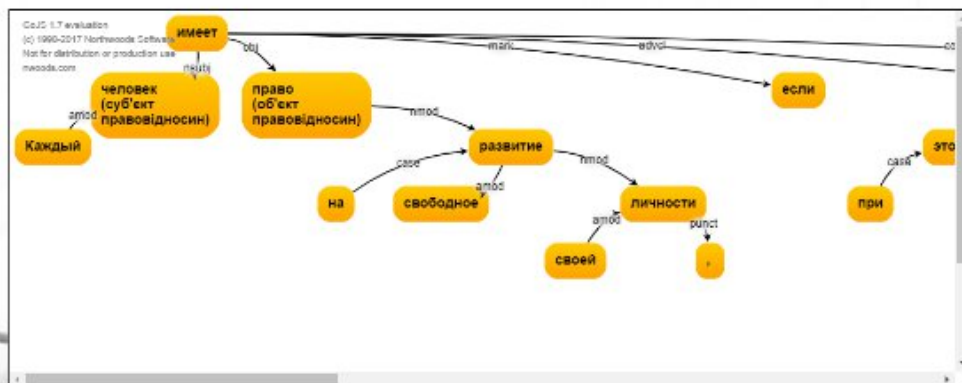


## Помилковий результат аналізу українського законодавчого тексту за допомогою російської мовної моделі



## Семантичний аналіз

*"Каждый человек имеет право на свободное развитие своей личности, если при этом не нарушаются права и свободы других людей, и имеет обязанности перед обществом, в котором обеспечивается свободное и всестороннее развитие его личности."*



## Можливе використання

- Створення морально-етичних запобіжників систем штучного інтелекту;
- Створення машинного коду шляхом опису його природною мовою;
- Створення алгоритмів смарт-контрактів без знання програмування;
- Створення інтелектуальних систем віртуальних помічників

**Дякую за увагу!**