УДК 004.822 https://doi.org/10.15407/pp2020.02-03.259

ЗАСТОСУВАННЯ БАЗ ЗНАНЬ ТА ДЕСКРИПТИВНИХ ЛОГІК ДО АНАЛІЗУ ПРИРОДНОМОВНИХ ТЕКСТІВ

Г.І. Гогерчак

У статті наведено деякі шляхи використання баз знань для аналізу природномовних текстів та розв'язання деяких задач їх обробки. Розглядаються основні задачі обробки природних мов, що є базою для їх семантичного аналізу: задачі токенізації, визначення частин мови, пошуку граматичних залежностей, пошуку кореферентностей. Подано основні поняття теорії баз знань та запропоновано підхід до їх наповнення на основі залежностей фреймворку Universal Dependencies та задачі кореферентності. Наведено приклади використання баз знань, наповнених з текстів природною мовою, для вирішення прикладних задач: перевірки змістовності побудованої синтаксичної та семантичної моделей, побудови відповідей на запитання.

Ключові слова: бази знань, обробка природних мов, синтаксичні залежності, пошук кореферентностей, семантичний аналіз.

В статье приведены пути использования баз знаний для анализа естественноязычных текстов и решения некоторых задач их обработки. Рассматриваются основные задачи обработки естественных языков, что являются базой для их семантического анализа: задачи токенизации, определение частей речи, поиска грамматических зависимостей, поиска кореферентностей. Представлены основные понятия теории баз знаний и предложен подход к их наполнения на основе зависимостей фреймворка Universal Dependencies и задачи кореферентности. Приведены примеры использования баз знаний, наполненных из текстов на естественном языке, для решения прикладных задач: проверки содержательности построенной синтаксической и семантической моделей, построения ответов на вопросы.

Ключевые слова: базы знаний, обработка естественных языков, синтаксические зависимости, поиск кореферентностей, семантический анализ.

The article describes some ways of knowledge bases application to natural language texts analysis and solving some of their processing tasks. The basic problems of natural language processing are considered, which are the basis for their semantic analysis: problems of tokenization, parts of speech tagging, dependency parsing, correference resolution. The basic concepts of knowledge bases theory are presented and the approach to their filling based on Universal Dependencies framework and the correference resolution problem is proposed. Examples of applications for knowledge bases filled with natural language texts in practical problems are given, including checking constructed syntactic and semantic models for consistency and question answering.

Key words: knowledge bases, natural language processing, syntax dependencies, coreference resolution, semantic analysis.

Вступ

Історія галузі обробки природних мов (natural language processing) загалом бере свій початок в 50-х роках XX століття. Першою перед науковцями постала задача автоматизації перекладу: для уряду Сполучених Штатів була важливою наявність системи, що б дозволяла здійснювати переклад російськомовних текстів англійською мовою з високою точністю. Вже 1954 року в рамках Джорджтаунського експерименту було вперше продемонстровано примітивну систему машинного перекладу.

Сучасна галузь природної обробки мов налічує понад три десятки основних задач, серед яких задачі визначення частин мови, токенізації тексту, пошуку залежностей, побудови синтаксичного дерева, пошуку кореферентностей, лексичної нормалізації, розпізнавання іменованих сутностей, пошуку пропущених компонент, перевірки природномовних висновків, виявлення відношень, розуміння мовлення, машинного перекладу, сентимент-аналізу, перевірки граматики тощо. Актуальний стан великої кількості цих та подібних задач описано на порталі NLP-progress[1] та на сторінці Natural Language Processing порталу Papers With Code[2], де зокрема подано рейтинги моделей розв'язання кожної із задач з посиланнями на наукові статті, що ці моделі описують, а також, для великої кількості моделей, – посилання на вихідні коди відповідних моделей машинного навчання.

Серед таких задач слід виокремити задачу виявлення відкритої інформації (open information extraction), мета якої – представлення природномовного тексту у структурованому вигляді: зазвичай у вигляді бінарних відношень чи відношень більших розмірностей. Якісне розв'язання цієї задачі дало б можливість говорити про наявність автоматизованих методів наповнення бази знань з природномовних даних, зміст яких і складається з атомарних концептів та ролей – відношень між ними.

На час написання цього тексту ця задача не має чітко сформульованих та загальноприйнятих стандартів результату, тобто які саме відношення повинні бути одержані та яким чином вони повинні бути оформлені. З огляду на це, відсутні також стандарт оцінювання моделей та корпуси прийнятного розміру для якісного навчання ML-моделей, як це прийнято для багатьох вище описаних задач області обробки природномовних текстів.

[1] http://nlpprogress.com/

[2] https://paperswithcode.com/area/natural-language-processing/

Перші кроки в напрямку специфікації та оцінки результатів цієї задачі були зроблені в [1], де пропонується порівняння OpenIE-моделей на основі кривої точність-повнота та метрики AUC (площа під кривою). Більшість нових моделей використовують для порівняння отриманих результатів саме запропоновану там методику оцінювання результатів [2], хоча деякі нові напрацювання пропонують порівняння на базі більш звичної метрики F1 [3, 4].

Загалом моделі, що пропонуються для розв'язання цієї задачі поділяють на два підтипи [5]:

- системи на базі машинного навчання (наприклад, Neural Open Information Extraction та OpenIE-5.0);
- системи на базі правил (наприклад, Graphene [2]).

Слід зазначити, що якість сучасних моделей для розв'язання цієї задачі (навіть вимірювана за допомогою існуючих метрик F1 та AUC) не дозволяє говорити про якісну побудову баз знань на основі природномовної текстової інформації на даному етапі, зокрема з огляду на різноманіття формулювань задачі та метрик для порівняння результатів.

Таким чином перспективним напрямком досліджень наразі є видобування відкритих (довільних) відношень з природномовних текстів, зокрема формалізація задачі OpenIE з огляду на її застосування в наповнені баз знань з природномовних текстів, побудова апарату метрик для порівняння моделей розв'язання задачі в заданій формальній системі та власне розв'язання поставленої задачі.

Побудова бази знань на основі природномовного тексту дає можливість здійснювати аналіз властивостей тексту за допомогою алгоритмів та методів роботи з базами знань на основі дескриптивних логік. Так, використання алгоритму перевірки виконуваності концептів за допомогою семантичного табло для побудованої на основі тексту бази знань дає змогу здійснювати перевірку змістовності (сумісності) побудованої синтаксичної та семантичної моделі. Таким чином, оперуючи певними додатковими знаннями про предметну область, можна ідентифікувати суперечливі, а отже помилкові елементи з метою подальшого їх виправлення. Частково розв'язна проблема виконання запитів до бази знань, у разі перетворення текстового запитання на відповідний вираз мови запитів, також є корисним засобом розв'язання задачі відповідей на запитання (question answering).

Огляд окремих задач обробки природних мов

Потенційно корисними вхідними даними для задачі видобування відкритих відношень можуть бути результати аналізу тексту на предмет частин мови, іменованих сутностей, граматичних залежностей та кореферентностей.

Задача токенізації. Задача токенізації в галузі обробки природної мови полягає у обробці послідовності символів (тексту) та виявлення в ній окремих слів або речень. Поділ послідовності на слова в першому наближенні може здійснюватися за рахунок розбиття вхідного потоку символів на частини за розділювачами (наприклад, пробілами, знаками пунктуації). Повноцінна ж токенізація повинна враховувати також особливості певних мов, де знаки пунктуації можуть бути частинами складних лексичних конструкцій (наприклад, в англійській мові послідовність символів *i. e.* відповідає українському словосполученню *іншими словами*, а конструкція *let's* позначає два слова: *let* та *us*) чи скорочень. Аналогічним чином не можна цілком звести до розбиття тексту за розділювачами й поділ на речення, адже знаки пунктуації, як це зазначено вище, можуть виступати в тому числі й частиною слів та складних мовленнєвих конструкцій.

Оскільки, як було зазначено вище, алгоритми токенізації враховують особливості мови, текст якої подається на вхід, алгоритми токенізації зазвичай будуються для кожної мови чи групи подібних мов окремо. Так, поділ англомовних текстів на слова та речення може здійснюватися за допомогою Stanford Tokenizer, запропонованого в [6].

Задача визначення частин мови. Задача визначення частин мови (POS tagging) полягає у позначенні кожного слова в тексті частиною мови, до якої це слово належить. Сучасні розробки у галузі обробки природних мов здебільшого користуються морфологічними позначками, визначеними в Universal Dependencies [7] – фреймворку для єдиної системи анотації граматики різних природних мов. Цей фреймворк дозволяє працювати з морфологічною та граматичною структурою речення, абстрагуючись від конкретної мови і оперуючи лише відповідними універсальними позначеннями.

Розглянемо наступне речення:

Усні повідомлення записують на папері, замінюючи звуки людської мови літерами алфавіту.

Відповідний йому результат морфологічного розбору у форматі Universal Dependencies показано на рис. 1.

Рис. 1. Результат визначення частин мови речення

Тут позначкою *ADJ* позначено прикметники, *NOUN* – іменники, *VERB* – дієслова, *ADP* – прийменники, *PUNCT* – знаки пунктуації.

Сучасні моделі визначення частин мови здебільшого засновані на підході машинного навчання. Так, для розв'язання цієї задачі для англійської мови зазвичай використовують стандартний набір даних – частину Penn Treebank, пов'язану з Wall Street Journal, що містить 45 різних POS-тегів. Найкращий показник точності на момент написання цього тексту на рівні 97.96 % демонструє модель Meta BiLSTM, запропонована в [8]. В основу цієї моделі покладено дві рекурентні нейронні мережі з контекстом на рівні речення, результати яких поєднуються за допомогою мета-моделі так, що на виході одержуються уніфіковані представлення кожного слова, які потім використовуються для позначень.

Розв'язання аналогічної задачі для багатьох мов одночасно, з використанням тегів з фреймворку Universal Dependencies та відповідних корпусів для різних мов – більш складна задача. На даний момент кілька моделей, зокрема Uppsala та HIT-SCIR демонструють кращі результати для великої кількості мов (середній показник F1 score за усіма мовами для обох моделей перевищує 0.9 для цієї задачі). Зокрема моделі HIT-SCIR та Stanford дають F1 score вище 0.97 для англійської, української та російської мов.

Задача пошуку залежностей. Задача пошуку залежностей (dependency parsing) полягає у виявленні в реченні залежностей, що представляють його граматичну структуру і визначають зв'язки між «основними» словами і словами, що їх модифікують.

Загальні принципи позначення синтаксичних залежностей також подано у фреймворку Universal Dependencies, який визначає понад 30 різних типів залежностей та деякі розширення для них залежно від групи мов, що розглядаються. В базовій версії цих залежностей синтаксична структура речення подається у вигляді дерева, тобто кожне слово речення (окрім головного – кореня) має рівно одного предка. Кожна гілка дерева помічена спеціальною позначкою, яка категоризує зв'язок між словом-предком та словом-нащадком за одним із 36 різних типів залежностей.

Приклад результату такого синтаксичного аналізу розглянутого вище речення показано на рис. 2.

Рис. 2. Результат пошуку залежностей

Тут позначкою *amod* позначено прикметникове означення, *obj* – об'єкт дії, *obl* – обставину, *advcl* – дієприслівниковий зворот, *case* – допоміжне слово, *nmod* – іменникове означення, *punct* – знаки пунктуації.

Моделі розв'язання цієї задачі для англійської мови здебільшого порівнюють на основі набору даних Penn Treebank з передбаченими позначеннями частин мови. Для їх порівняння використовуються наступні метрики:

- UAS (unlabeled attachment score), що не бере до уваги позначки залежностей, а порівнює лише коректність визначення предка кожного слова;
- LAS (labeled attachment score), що позначає частку коректно розібраних слів (правильно визначених як предків, так і позначок залежностей).

На час написання цього тексту найкращі значення наведених вище метрик демонструє модель Label Attention Layer + HPSG + XLNet, запропонована в листопаді 2019 року в [9]. Ця модель також базується на нейромережевому підході та досягає UAS 97.33 % і LAS 96.29 %. Проте останні наукові конференції зосереджені на побудові єдиних моделей синтаксичного розбору для великої кількості мов. Так, модель HIT-SCIR дозволяє досягти LAS у 0.92, 0.88 та 0.87 для російської, української та англійської мов відповідно.

Розглянемо більш складне речення: *Коти зазвичай ловлять мишей та щурів , живляться ними.*

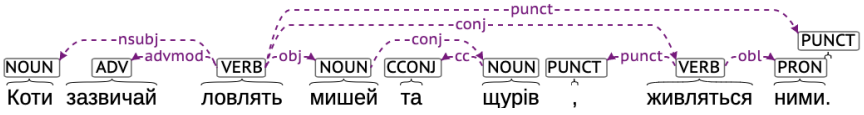

Рис. 3. Результат пошуку залежностей

Зі вказаного вище дерева залежностей можна помітити, що базовий набір залежностей не дозволяє в достатній мірі здійснювати аналіз синтаксичних зв'язків та видобування з них семантичних. Так, сурядні об'єкти дії *мишей* та *щурів* тут пов'язані зв'язком кон'юнкції *conj*, через що слово *щурів* виявляється пов'язаним з дією *ловлять* лише опосередковано, хоча семантично теж є її об'єктом.

Цю та інші проблеми вирішують шляхом розширення дерева залежностей додатковими дугами (рис. 4) – звісна річ, зі втратою деревовидної структури. Перетворення базового дерева залежностей в розширений граф залежностей потребує зокрема вирішення наступних проблем:

- відновлення випущених слів шляхом створення фіктивних токенів;
- поширення зв'язків (об'єктів, суб'єктів, означень) через кон'юнкцію;
- поширення суб'єктів на підлеглі дієслова складного предиката;
- обробка підрядного речення, що уточнює певний об'єкт, як дії виконаної цим об'єктом (може призводити до утворення циклів);
- додавання допоміжного слова в назву залежності.

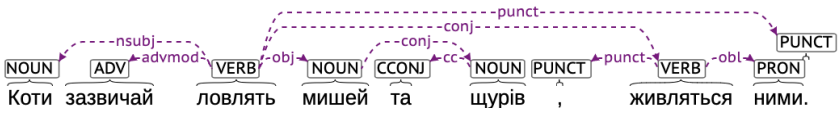

Рис. 4. Розширений граф синтаксичних залежностей

Пакет обробки природних мов CoreNLP дозволяє досягти значення 0.92 для показника LAS для англійської мови[3]. Наявність корпусу для української дозволяє говорити про потенційну можливість розв'язання цієї задачі також і для цієї мови, проте у відкритому доступі моделей для неї на час написання цього тексту не виявлено.

Поряд з універсальними залежностями існує також низка спеціалізованих під конкретні мови форматів залежності. Приклад альтернативного формату опису синтаксичної структури речення українською мовою, запропонованого у [10], подано у порівнянні з універсальними залежностями на рис. 5, 6.

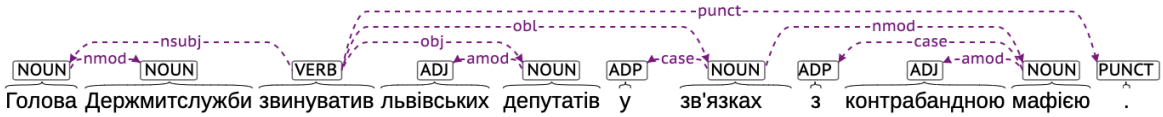

Рис. 5. Приклад універсального дерева залежностей для україномовного речення

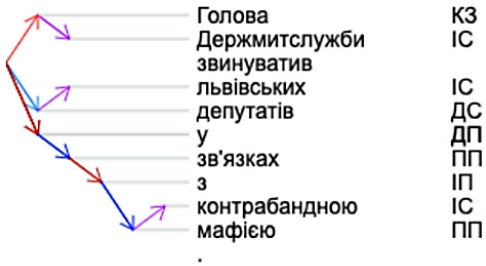

Рис. 6. Приклад альтернативного дерева залежностей для україномовного речення

Тут позначкою *КЗ* позначено сполуку підмета й присудка, *ІС* – іменникову безприйменникову сполуку, *ДС* – дієслівну безприйменникову сполуку, *ДП* – дієслівну прийменникову сполуку, *ПП* – прийменникову сполуку, *ІП* – іменникову прийменникову сполуку.

Комбінування результатів різних форматів синтаксичного розбору дозволяє досягти більш якісного агрегованого результату та здійснювати корекцію помилок, що виникли в кожному з отриманих дерев.

Задача пошуку кореферентностей. Задача пошуку кореферентностей (coreference resolution) полягає у кластеризації згадувань в тексті, що стосуються однієї й тієї ж сутності реального світу.

Розглянемо наступне речення: *«Я голосувала за Барака Обаму, оскільки його переконання найближчі до моїх власних цінностей» – мовила вона.*

Аналіз його синтаксичних залежностей (рис. 7) не дозволяє в повній мірі визначити, яких об'єктів реального світу: однакових чи різних, – стосуються подані в реченні займенники. Аналогічну проблему, але розширену на згадки в усьому тексті, а не лише в межах речення, взагалі не вдасться вирішити за допомогою ані базових, ані розширених залежностей, адже вони представляють зв'язки лише в межах одного речення.

[3] https://nlp.stanford.edu/pubs/schuster2016enhanced.pdf

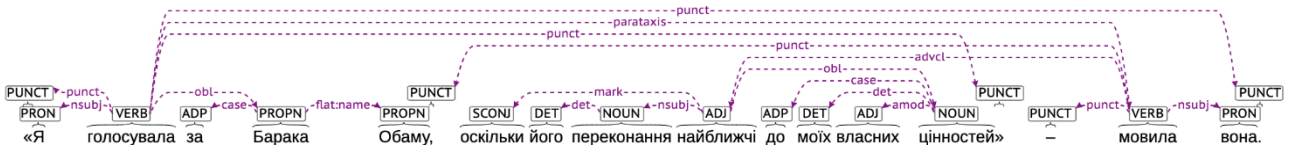

Рис. 7. Дерево залежностей речення з багатьма кореферентними словами та словосполученнями

Вищесказане обумовлює появу окремої задачі обробки природних мов, вирішення якої б дозволило зібрати еквівалентні сутності в певному тексті в одну та аналізувати всі відношення для неї більш повно.

Множина кореферентних слів та словосполучень зазвичай подається у вигляді лісу (рис. 8) – множини дерев, кожне з яких позначає собою множину кореферентних вузлів. Дуга кореферентності зазвичай спрямовується до найбільш конкретного позначення об'єкта реального світу.

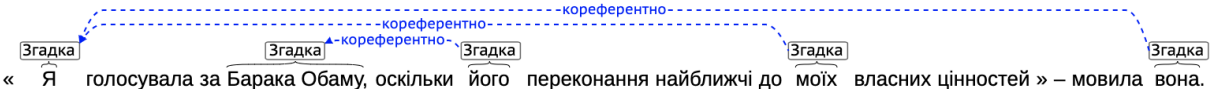

Рис. 8. Дерево кореферентностей

Порівняння моделей пошуку кореферентностей наразі здійснюється на корпусі OntoNotes[4], що містить англомовні тексти різних жанрів (новини, розмовна телефонна мова, блоги, ток-шоу тощо) із зокрема проанотованими кореферентностями.

Наразі найкращі показники для цієї задачі показують модифікації моделі BERT [11] на базі підходу машинного навчання, розробленої командою Google AI Language. BERT пропонує спільну модель подання природномовної інформації для сукупності задач обробки текстів та запроваджує врахування контексту з обох боків від слова, на відміну від використання лівосторонніх чи правосторонніх контекстів у попередніх ефективних моделях.

Основні поняття теорії баз знань

Базові засади онтологоподібних систем та баз знань природномовних текстів подані в роботах [12, 13, 14, 15, 16]. Введемо деякі поняття теорії баз знань, що використовуватимуться далі.

Концепти є інструментом для запису знань про предметну область, до якої вони відносяться. Ці знання поділяються на загальні знання про поняття і їх взаємозв'язки та знання про індивідуальні об'єкти, їх властивості і зв'язки з іншими об'єктами. Відповідно до цього поділу знання, що записуються за допомогою мови дескриптивних логік, поділяються на множину термінальних аксіом TBox і множину фактів про індивідів ABox.

Означення 1. Термінологічною аксіомою називається вираз $C \sqsubseteq D$ (включення концепта C в концепт D) або $C \equiv D$ (еквівалентність концептів C і D), де C і D – довільні концепти.

Означення 2. Термінологією (TBox) називається довільна скінченна множина термінологічних аксіом.

Означення 3. Аксіома $C \sqsubseteq D$ ($C \equiv D$) істинна в інтерпретації I, якщо $C^I \subseteq D^I$ ($C^I = D^I$). I в даному випадку називається моделлю цієї аксіоми і пишуть $I \vDash C \sqsubseteq D$. Інтерпретацію I називають моделлю термінології T і пишуть $I \vDash T$, якщо I є моделлю для всіх аксіом із T.

Означення 4. Термінологія називається сумісною або виконуваною, якщо вона має непусту модель. Концепт C виконується відносно термінології T, якщо існує модель I термінології T така, що $C^I \neq \varnothing$.

Термінологія дає можливість записувати загальні знання про концепти і ролі. Але часто інсує необхідність записувати знання про конкретні індивіди: якому класу належить індивід, якими відношеннями (ролями) вони зв'язані одне з одним.

Означення 5. Системою фактів (ABox) називається скінченна множина A тверджень виду $a:C$ та aRb, де a,b є індивідами, C – довільний концепт, а R – роль.

Підхід до наповнення бази знань на основі синтаксичних залежностей

Деякі підходи до аналізу природної мови на предмет видобування знань представлені раніше в роботах [17, 18]. Нижче розглядається концептуально відмінний підхід до наповнення бази знань з використанням універсальних залежностей та кореферентностей.

[4] https://catalog.ldc.upenn.edu/LDC2013T19

Дерево (або граф) синтаксичних залежностей, розглянутий вище, є потужним джерелом для видобування з нього знань у вигляді відкритих відношень. Розглянемо наступний текст:

Фільм "Ла-Ла Ленд" – третя робота молодого режисера Дем'єна Шазелла. Попередня його картина "Одержимість" здобула чимало найпрестижніших кінопремій, в тому числі одразу три премії "Оскар". Сьогодні вже відомі цьогорічні номінанти, і стрічка "Ла-Ла Ленд" є беззаперечним лідером – має 14 номінацій. Вона вже перемогла у всіх найпрестижніших номінаціях премії "Золотий глобус".

Основними джерелами відношень, тобто фактів вигляду aRb, є дієслова разом зі словами, що пов'язані з ними залежностями *nsubj* (іменниковий суб'єкт) та *obj* (об'єкт). Розглянемо друге речення вказаного вище тексту, дерево залежностей для якого подано на рис. 9.

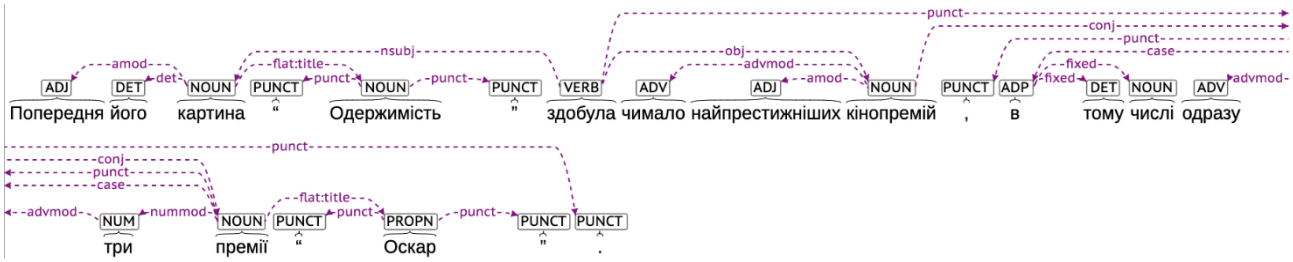

Рис. 9. Дерево залежностей

З цього дерева залежностей очевидним чином можна вилучити відношення (*картина*; *здобути*; *кінопремія*). Очевидно, що таке відношення саме по собі не несе достатнього змістового навантаження – все через те, що індивіди, які беруть участь в цьому відношенні, потребують достатньої конкретизації.

Нехай суб'єктові цієї дії відповідає індивід a_1, а об'єктові – індивід a_2. Використовуючи залежності *flat* та *amod* сконструюємо наступних ланцюг концептів для індивіда a_1:

$$картина\_попередня \sqsubseteq картина, картина\_одержимість \sqsubseteq картина\_попередня$$

Факт приналежності індивіда згенерованим концептам можна записати так: $a_1 : картина\_одержимість$. Слід зазначити, що приналежність цього індивіда двом іншим концептам випливає зі змісту відношення включення концептів.

Аналогічним чином з індивіда a_2 TBox поповнюється наступними термінологічною аксіомою:

$$кінопремія\_престижна \sqsubseteq кінопремія$$

ABox, відповідно, поповниться фактом $a_2 : кінопремія\_престижна$, а також фактом $a_1 R_{здобути} a_2$. Оскільки об'єкт в реченні подано у множині, слід передбачити наступне вкладення концептів:

$$картина\_одержимість \sqsubseteq \geq 2R_{здобути} . кінопремія\_престижна$$

Ще один суб'єкт ролі $R_{здобути}$ приховано в базовому дереві залежностей за зв'язком сурядності *conj*. Після виконання аналогічних операцій і для нього отримаємо таку базу знань:

$$\text{TBox} = \{ картина\_попередня \sqsubseteq картина, картина\_одержимість \sqsubseteq картина\_попередня,$$

$$кінопремія\_престижна \sqsubseteq кінопремія, премія\_оскар \sqsubseteq премія,$$

$$картина\_одержимість \sqsubseteq \geq 2R_{здобути} . кінопремія\_престижна,$$

$$картина\_одержимість \sqsubseteq = 3R_{здобути} . премія\_оскар \}$$

$$\text{ABox} = \{ a_1 : картина\_одержимість, a_2 : кінопремія\_престижна, a_3^1 : премія\_оскар, a_3^2 : премія\_оскар,$$

$$a_3^3 : премія\_оскар, a_1 R_{здобути} a_2, a_1 R_{здобути} a_3^1, a_1 R_{здобути} a_3^2, a_1 R_{здобути} a_3^3 \}$$

Окрім залежностей *flat*, *nmod* та *amod* нові термінологічні аксіоми можуть утворюватися й із залежностей типу *obj* у випадку випущеного присудка. Так, у першому реченні наведеного вище тексту (рис. 10) випущено дієслово *є*.

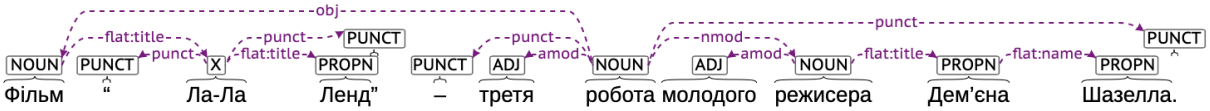

Рис. 10. Дерево залежностей

Таким чином, оскільки в корені дерева залежностей знаходиться іменник, залежності *obj* та *subj* тут семантично позначають вкладення концептів. Аналогічним чином продукуємо концепти та термінологічні аксіоми для об’єкту та кореня:

фільм_Ла-Ла_Ленд ⊑ *фільм*, *робота_режисера* ⊑ *робота*,

робота_режисера_молодого ⊑ *робота_режисера*,

робота_режисера_Дем’єна_Шазелла ⊑ *робота_режисера_молодого*,

робота_режисера_Дем’єна_Шазелла_третя ⊑ *робота_режисера_Дем’єна_Шазелла*

Додатково для наведеного вище речення до переліку термінологічних аксіом буде додано наступну:

фільм_Ла-Ла_Ленд ⊑ *робота_режисера_Дем’єна_Шазелла_третя*

В тих випадках, коли коренем речення є прикметник, теж відбувається вкладення концептів. Так, перша частина третього речення (рис. 11) продукує наступні термінологічні аксіоми:

цьогорічний_номінант ⊑ *номінант*, *цьогорічний_номінант* ⊑ *відомий*

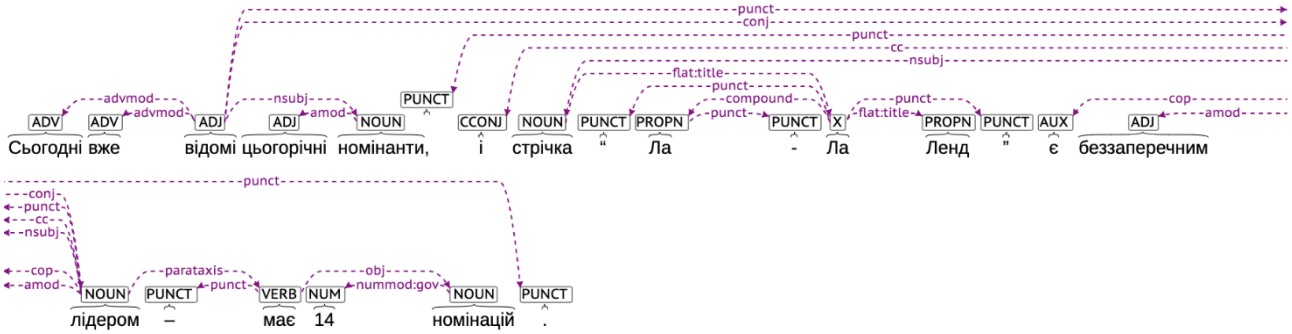

Рис. 11. Дерево залежностей

У випадках, коли корінь речення має сурядні, тобто пов’язані з ним залежністю *conj*, елементи, здійснюється аналогічний розгляд кожного такого сурядного компонента як кореня свого піддерева. Оскільки коренем піддерева є іменник, аналогічно до розглянутого раніше випадку, база знань поповнюється наступними термінологічними аксіомами:

стрічка_Ла-Ла_Ленд ⊑ *стрічка*, *лідер_беззаперечний* ⊑ *лідер*,

стрічка_Ла-Ла_Ленд ⊑ *лідер_беззаперечний*

Аналогічним чином слід обробляти й піддерева, що залежать від кореня зв’язком *parataxis*. Так, в даному реченні в якості окремого твердження ми розглядаємо вираз *має 14 номінацій*, який додає до бази знань лише концепт *номінація*. Оскільки дієслово *має* не має суб’єкта, виникає проблема його визначення з контексту. В цьому випадку визначення зробити доволі просто: достатньо використати суб’єкт твердження-предка, тобто *стрічка «Ла-Ла Ленд»*. Таким чином додатково до бази знань слід додати наступні факти:

$$a_4 : \textit{стрічка\_Ла-Ла\_Ленд},\ a_5 : \textit{номінація},\ a_4 R_{\textit{мати}} a_5$$

Останнє речення (рис. 12) продукує наступні термінологічні аксіоми:

номінація_премії ⊑ *номінація*, *номінація_премії_Золотий_глобус* ⊑ *номінація_премії*,

номінація_премії_Золотий_глобус_найпрестижніша ⊑ *номінація_премії*

Якщо вважати задачу пошуку кореферентностей вирішеною, то суб'єкт *вона* відповідає концепту *стрічка_Ла-Ла_Ленд*. Таким чином, база знань також доповнюється такими фактами:

$$a_6 : \text{номінація\_премії\_Золотий\_глобус\_найпрестижніша},\ a_4 R_{\text{перемогти\_у}} a_6$$

Остаточна база даних розглянутого фрагменту тексту матиме наступний вигляд:

CN = { *картина_попередня, картина, картина_одержимість, кінопремія_престижна, кінопремія, премія_оскар, премія, фільм_Ла-Ла_Ленд, фільм, робота_режисера, робота, робота_режисера_молодого, робота_режисера_Дем'єна_Шазелла, робота_режисера_Дем'єна_Шазелла_третя, фільм_Ла-Ла_Ленд, цьогорічний_номінант, номінант, відомий, стрічка_Ла-Ла_Ленд, стрічка, лідер_беззаперечний, лідер, стрічка_Ла-Ла_Ленд* }

RN = { $R_{\text{здобути}}$, $R_{\text{мати}}$ } IN = { $a_1, a_2, a_3^1, a_3^2, a_3^3, a_4, a_5$ }

TBox = { *картина_попередня* $\sqsubseteq$ *картина, картина_одержимість* $\sqsubseteq$ *картина_попередня, кінопремія_престижна* $\sqsubseteq$ *кінопремія, премія_оскар* $\sqsubseteq$ *премія,*

картина_одержимість $\sqsubseteq \geq 2 R_{\text{здобути}}$. *кінопремія_престижна,*

картина_одержимість $\sqsubseteq = 3 R_{\text{здобути}}$.*премія_оскар, фільм_Ла-Ла_Ленд* $\sqsubseteq$ *фільм,*

робота_режисера $\sqsubseteq$ *робота, робота_режисера_молодого* $\sqsubseteq$ *робота_режисера*

робота_режисера_Дем'єна_Шазелла $\sqsubseteq$ *робота_режисера_молодого,*

робота_режисера_Дем'єна_Шазелла_третя $\sqsubseteq$ *робота_режисера_Дем'єна_Шазелла,*

фільм_Ла-Ла_Ленд $\sqsubseteq$ *робота_режисера_Дем'єна_Шазелла_третя,*

цьогорічний_номінант $\sqsubseteq$ *номінант, цьогорічний_номінант* $\sqsubseteq$ *відомий, стрічка_Ла-Ла_Ленд* $\sqsubseteq$ *стрічка,*

лідер_беззаперечний $\sqsubseteq$ *лідер, стрічка_Ла-Ла_Ленд* $\sqsubseteq$ *лідер_беззаперечний, номінація_премії* $\sqsubseteq$ *номінація,*

номінація_премії_Золотий_глобус $\sqsubseteq$ *номінація_премії,*

номінація_премії_Золотий_глобус_найпрестижніша $\sqsubseteq$ *номінація_премії* }

ABox = { a_1 *: картина_одержимість,* a_2 *:кінопремія_престижна,* a_3^1 *:премія_оскар,* a_3^2 *:премія_оскар,*

a_3^3 *:премія_оскар,* a_4 *: стрічка_Ла-Ла_Ленд,* a_5 *: номінація,*

a_6 *: номінація_премії_Золотий_глобус_найпрестижніша,*

$a_1 R_{\text{здобути}} a_2$, $a_1 R_{\text{здобути}} a_3^1$, $a_1 R_{\text{здобути}} a_3^2$, $a_1 R_{\text{здобути}} a_3^3$, $a_4 R_{\text{мати}} a_5$, $a_4 R_{\text{перемогти\_у}} a_6$ }

Використовуючи під час семантичного аналізу лексичної бази WordNet, можна також додати наступні допоміжні термінологічні аксіоми:

{ фільм $\sqsubseteq$ *картина, картина* $\sqsubseteq$ *робота, кінопремія* $\sqsubseteq$ *премія, фільм* $\equiv$ *стрічка,*

фільм_Ла-Ла_Ленд $\equiv$ *стрічка_Ла-Ла_Ленд }*

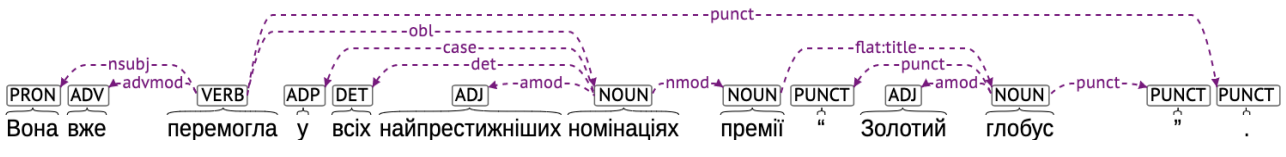

Рис. 12. Дерево залежностей

Пошук змістових суперечностей за допомогою алгоритму семантичного табло

Означення 6. Алгоритм U розв'язує проблему виконуваності концептів в термінології T для дескриптивної логіки L, якщо виконуються наступні умови:

1) термінальність: для довільних (C,T) алгоритм U генерує відповідь $U(C,T)$ в скінченному часі;

2) коректність: для довільних (C,T), якщо C виконується в термінології T, то $U(C,T)=1$;

3) повнота: для довільних (C,T), якщо $U(C,T)=1$, то концепт C виконується в термінології T.

Алгоритм семантичного табло для перевірки виконуваності концептів задається правилами в таблиці.

Таблиця. Правила алгоритму семантичного табло для логіки $\mathcal{ALC}$ +TBox

Правило	**Умови застосування**	**Дія**
$\sqcap$-правило	точка x – активна; $x:(C\sqcap D)\in\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{x:C,x:D\}$
$\sqcup$-правило	точка x – активна; $x:(C\sqcup D)\in\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{x:C\},\mathcal{A}''=\mathcal{A}\cup\{x:D\}$
$\exists$-правило	точка x – активна; $x:\exists R.C\in\mathcal{A}$; $\nexists y:\{xRy,y:C\}\subseteq\mathcal{A}$	y – нащадок x; $\mathcal{A}'=\mathcal{A}\cup\{xRy,y:C\}$
$\forall$-правило	точка x – активна; $x:\forall R.C\in\mathcal{A}$; $\exists y:xRy\in\mathcal{A}\wedge y:C\notin\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{y:C\}$
T-правило	точка x – активна; $x:E\notin\mathcal{A}$, де $E\sqsubseteq\top\in T$	$\mathcal{A}'=\mathcal{A}\cup\{x:E\}$

З початкового ABox $\mathcal{A}_0$ шляхом застосування описаних правил буде побудовано дерево пошуку, в якому в корені знаходиться $\mathcal{A}_0$ і у кожного ABox є 0, 1 або 2 нащадка. Застосування правил припиняється, якщо до чергового ABox $\mathcal{A}$ не застосовне жодне з правил, або якщо в $\mathcal{A}$ міститься явна суперечність (тобто для деякого індивіда x та концепта C $\{x:C,x:\neg C\}\subseteq\mathcal{A}$, або ж $\{x:\bot\}\subseteq\mathcal{A}$).

Для виконання умови термінальності алгоритму, вводиться поняття активної точки, аби $\exists$-правило та T-правило разом не призводили до нескінченного генерування індивідів з однаковим набором концептів, до яких вони належать.

Означення 7. Точка x блокує точку y, якщо x – предок y і $L(x)\supseteq L(y)$, де $L(x)=\{C|x:C\in\mathcal{A}\}$. Точка y називається заблокованою, якщо її блокує деяка точка x. Активною називається точка, що не є ані заблокованою, ані нащадком деякої заблокованої точки.

Алгоритм семантичного табло для розглянутого вище тексту не призводить до суперечностей, що означає його змістовність та несуперечність.

З іншого боку, якщо доповнити базу знань фактом *картина_одержимість* $\sqcap$ $R_{здобути}.премія\_оскар\sqsubseteq\bot$, який означає *картина «Одержимість» не здобула жодної премії «Оскар»*, алгоритм зупиниться на явній суперечності $a_1:\bot$.

Побудова відповідей на запитання до тексту

Для формулювання запитів введемо новий сорт символів – скінченну мнижину індивідних змінних $Var=\{x_0,x_1,\ldots\}$. Атомарним запитом називатимемо вирази вигляду $u:C$ та uRv, де C – концепт, R – роль, $u,v\in IN\cup Var$.

Означення 8. Кон'юнктивний запит – це вираз виду $\exists\bar{v}(t_1\wedge\ldots\wedge t_k)$, де t_i – атоми, $\bar{v}=\{v_1,\ldots,v_l\}$ – список деяких змінних, що входять в t_i. Змінні v_i називаються зв'язаними, а решта змінних – вільними. Якщо $\bar{v}=\{v_1,\ldots,v_l\}$ – перелік вільних змінних запиту q, записуватимемо це як $q(\bar{v})$.

Розглянемо вказану вище базу знань. Природномовний запит *які стрічки здобули премію Оскар*? може бути записаний у вигляді наступного запиту: $q(x)=x:\exists R_{здобути}.премія\_оскар$. Відповідь на запит – множина індивідів, що задовільняють вказаним умовам. Для вищенаведеного прикладу відповідь може подаватися у вигляді $\{a_1,a_4\}$. Слід зазначити, що теорія баз знань базується на переконанні про відкритість світу: база знань

представляє собою сукупність всіх моделей, в яких задані в ній аксіоми справедливі. А тому відповідь на запит до бази знань завжди є підмножиною повної відповіді на поставлене природномовне запитання, на відміну до запиту до бази даних, який завжди є точною множиною повної відповіді на запитання.

Висновки

Наявний стан розв'язання проблем обробки природних мов надає якісні вхідні дані для задачі наповнення баз знань з текстів природної мови. Так, дерево залежностей, побудоване згідно з фреймворком Universal Dependencies, дозволяє викокремлювати термінологічні аксіоми та факти бази знань, в тому числі з числовими обмеженнями. Проте, невирішеність проблеми пошуку кореферентностей для української мови не дозволяє говорити про достатньо якісний стан розв'язання задачі наповнення баз знань україномовними текстами, що підтверджує необхідність роботи над корпусом кореферентностей для української мови.

Розглянутий вище підхід до наповнення баз знань може бути розширений на випадки умовних речень, причинно-наслідкових виразів та адаптований до різного часового контексту тверджень, поданих у тексті. Відповідно, аналіз баз знань, що містять таку інформацію, потребує використання розширеного апарату дескриптивних логік, включаючи їх комбінацію з темпоральними логіками та використання додаткової системи фактологічних аксіом.

Література

1. Stanovsky G., Dagan I. Creating a Large Benchmark for Open Information Extraction. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, US. 2016. P. 2300–2305.
2. Cetto M., Niklaus C., Freitas A., Handschuh S. Graphene: A Context-Preserving Open Information Extraction System. *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Santa Fe, New Mexico. 2018. P. 94–98.
3. Zhan J., Zhao H. Span Based Open Information Extraction. 2019.
4. Léchelle W., Gotti F., Langlais P. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. 2019. P. 6–15.
5. Niklaus C., Cetto M., Freitas A., Handschuh S. A Survey on Open Information Extraction. *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA. 2018. P. 3866–3878.
6. Manning C., Grow T., Grenager T., Finkel J., Bauer J. Stanford Tokenizer. 2002.
7. McDonald R., Nivre J., Quirmbach-Brundage Y., Goldberg Y., Das D., Ganchev K., Hall K., Petrov S., Zhang H., Täckström O., Bedini C., Castelló N.B. and Lee J. Universal Dependency Annotation for Multilingual Parsing. *In Proceedings of ACL*. 2003. P. 92–97.
8. Bohnet B., McDonald R., Simões G., Andor D., Pitler E. and Maynez J. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018. P. 2642–2652.
9. Mrini K., Dernoncourt F., Bui T., Chang W. and Nakashole N. Rethinking Self-Attention: An Interpretable Self-Attentive Encoder-Decoder Parser. 2019.
10. Дарчук Н. Автоматичний синтаксичний аналіз текстів корпусу української мови. Українське мовознавство. 2013. № 43. С. 11–19.
11. Devlin J., Chang M.-W., Lee K. and Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT. 2019. P. 4171–4186.
12. Baader F., Calvanese D., McGuinness D., Nardi D. and Patel-Schneider P. The Description Logic Handbook. Cambridge University Press. 2007. P. 578.
13. Сергієнко І.В., Кривий С.Л., Провотар О.І. Алгебраїчні аспекти інформаційних технологій. К: Наукова думка. 2011. 399 с.
14. Кривий С.Л., Дарчук Н.П., Провотар О.І. Онтологоподібні системи аналізу природномовних текстів. Проблеми програмування. 2018. № 2-3. С. 132–139.
15. Крывый С.Л., Дарчук Н.П., Ясенова И.С., Головина А.Л., Соляр А.С. Методы и средства систем представления знаний. Publisher: ITHEA. *Inter. journ. «Information Content and Processing»*. 2017. Vol. 4. № 1. С. 62–99.
16. Палагин А.В., Крывый С.Л., Петренко Н.Г. Знание-ориентированные информационные системы с обработкой естественно-языковых объектов: основы методологии и архитектурно- структурная организация. *УСиМ*. 2009. № 3. С. 42–55.
17. Палагин А.В., Крывый С.Л., Петренко Н.Г. Об автоматизации процесса извлечения знаний из естественно-языковых текстов. Natural and Artificial Intelligence Intern. Book Series. Inteligent Processing. ITHEA. Sofia. 2012. N 9. P. 44–52.
18. Палагин А.В., Крывый С.Л., Бибиков Д.С. Обработка предложений естественного языка с использованием словарей и частоты появления слов. Natural and Artificial Intelligence Intern. Book Series. *Inteligent Processing*. ITHEA. Sofia. N 9. 2010. P. 44–52.

References

1. Stanovsky G., Dagan I. Creating a Large Benchmark for Open Information Extraction. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, US. 2016. P. 2300–2305.
2. Cetto M., Niklaus C., Freitas A., Handschuh S. Graphene: A Context-Preserving Open Information Extraction System. *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Santa Fe, New Mexico. 2018. P. 94–98.
3. Zhan J., Zhao H. Span Based Open Information Extraction. 2019.
4. Léchelle W., Gotti F., Langlais P. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. 2019. P. 6–15.
5. Niklaus C., Cetto M., Freitas A., Handschuh S. A Survey on Open Information Extraction. *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA. 2018. P. 3866–3878.
6. Manning C., Grow T., Grenager T., Finkel J., Bauer J. Stanford Tokenizer. 2002.
7. McDonald R., Nivre J., Quirmbach-Brundage Y., Goldberg Y., Das D., Ganchev K., Hall K., Petrov S., Zhang H., Täckström O., Bedini C., Castelló N.B. and Lee J. Universal Dependency Annotation for Multilingual Parsing. *In Proceedings of ACL*. 2003. P. 92–97.

8. Bohnet B., McDonald R., Simões G., Andor D., Pitler E. and Maynez J. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018. P. 2642–2652.
9. Mrini K., Dernoncourt F., Bui T., Chang W. and Nakashole N. Rethinking Self-Attention: An Interpretable Self-Attentive Encoder-Decoder Parser. 2019.
10. Darchuk N. Automated syntax analysis of texts from Ukrainian language corpus. Ukrainian linguistics. 2013. N 43. P. 11–19. (In Ukrainian)
11. Devlin J., Chang M.-W., Lee K. and Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT. 2019. P. 4171–4186.
12. Baader F., Calvanese D., McGuinness D., Nardi D. and Patel-Schneider P. The Description Logic Handbook. Cambridge University Press. 2007. P. 578.
13. Serhiienko I., Kryvyi S., Provotar O. Algebraic aspects of information technology. Scientific thought. 2011. 399 p. (In Ukrainian)
14. Kryvyi S., Darchuk N., Provotar O. Onlogoly-based systems of natural language analysis. *Problems of programming*. 2018. N 2-3. P. 132–139. (In Ukrainian)
15. Kryvyi S., Darchuk N., Yasenova I., Holovina O., Soliar A. Methods and means of knowledge representation systems. – Publisher: ITHEA. – *Inter. journ. «Information Content and Processing»*. 2017. Vol. 4. N 1. P. 62–99. (In Russian)
16. Palagin O., Kryvyi S., Petrenko N.. Knowledge-oriented information systems with the processing of natural language objects: the basis of ethodology, architectural and structural organization. Control Systems and Computers. 2009. N 3. P. 42–55. (In Russian)
17. Palagin O., Kryvyi S., Petrenko N. On the automation of the process of extracting knowledge from natural language texts. Natural and Artificial Intelligence Intern. Book Series. *Inteligent Processing.* ITHEA. Sofia. N 9. 2012. P. 44–52. (In Russian)
18. Palagin O., Kryvyi S., Bibikov D.. Processing natural language sentences using dictionaries and words frequency. Natural and Artificial Intelligence Intern. Book Series. *Inteligent Processing*. ITHEA. Sofia. N 9. 2010. P. 44–52. (In Russian)

Одержано 17.02.2020

Про автора:

Гогерчак Григорій Іванович,
аспірант І року навчання.
Кількість публікацій в українських виданнях – 4.
https://orcid.org/0000-0002-6898-2536.

Місце роботи автора:

Факультет комп'ютерних наук та кібернетики,
Київського національного університету імені Тараса Шевченка,
03680, Україна, м. Київ,
проспект Академіка Глушкова, 4д.
E-mail: gogerchak.g@gmail.com