

# АЛГОРИТМ ПОШУКУ КОРЕФЕРЕНТНИХ ОБ'ЄКТІВ В УКРАЇНОМОВНИХ ТЕКСТАХ ІЗ ВИКОРИСТАННЯМ ДЕРЕВ РІШЕНЬ

Сергій Погорілий, Павло Білецький

В роботі розглядається проблема пошуку кореферентних об'єктів в україномовних текстах використовуючи дерева рішень. Розроблено застосунок, що використовує векторні представлення слів Elmo та інші характеристики для автоматизованого формування дерева рішень. Для формування дерева та оцінки точності роботи алгоритму використано набір підготовлених текстів, що вміщає понад 360000 слів. Дерево рішень, створене для визначення, чи пара об'єктів є кореферентними, використано для формування кластерів кореферентних об'єктів. Використано спеціальні метрики для порівняння з результатами, отриманих іншими алгоритмами для української мови.

Ключові слова: кореферентність, обробка природних мов, дерева рішень, штучний інтелект, векторні представлення слів, нейронні мережі.

The paper examines the problem of coreference resolution in Ukrainian-language texts using decision trees. An application that uses vector representations of Elmo words and other characteristics for the automated formation of a decision tree has been developed. A set of prepared texts containing more than 360,000 words was used to form the decision tree and evaluate the accuracy of the algorithm. The decision tree created to determine whether a pair of objects is coreference was used to form clusters of coreference objects. Special metrics were used for comparison with the results obtained by other algorithms in the Ukrainian language.

Keywords: coreference resolution, natural language processing (NLP), decision trees, artificial intelligence (AI), vector words representation, neural networks.

## Вступ

Обробка природних мов (natural language processing, NLP) – велика галузь, що включає безліч задач: переклад між природними мовами, інтерфейси людина-комп'ютер, аналіз та генерація природного мовлення, виділення інформації з текстів. Однією із задач обробки природних мов є пошук кореферентних об'єктів у текстах. Кореферентність у текстах розуміється як відношення між синтаксичними одиницями, що вказують на один і той самий об'єкт (референт) в даному контексті [1].

Далі наведені приклади кореферентності (референт виділений жирним шрифтом, займенники - підкреслені).

Приклад простої анафори:

*Він перейшов через **гору**. Вона була високою.*

Приклад простої катафори:

*Вона вийшла на дорогу, що вела праворуч. **Марія** сьогодні мала гарний настрій.*

Приклад складеного антецеденту.

***Іван, Михайло та Остап** – всі вони працювали під землею.*

У порівнянні з іншими мовами Європи – англійською, німецькою, французькою, італійською, іспанською – для української мови властивий довільний порядок слів, як і для інших слов'янських мов – польської, російської, сербської, хорватської. До прикладу, для української мови:

*Карпо прикинув таке слівце, що **батько** перестав стругати і почав прислухатись. Він глянув на синів через хворостяну стіну. Сини стояли без діла й балакали, поспиравшись на заступи (Іван Нечуй-Левицький, «Кайдашева сім'я», оригінальний порядок слів).*

Із другого речення в прикладі можна перестановкою слів утворити інші граматично коректні речення з дуже близькими значеннями, але різними наголосами, залежно від задуму автора:

- Через хворостяну стіну він глянув на синів.

- На синів він глянув через хворостяну стіну.

- Через стіну хворостяну він глянув на синів.

- Через хворостяну стіну глянув він на синів.

- Через хворостяну стіну на синів глянув він.

Водночас в англійській мові можлива тільки одна комбінація: «He looked at his sons through the twig wall», оскільки в ній використовується стандартний порядок слів суб'єкт – дієслово – об'єкт (SVO, subject verb object). В інших мовах також поширений порядок суб'єкт – об'єкт – дієслово (SOV, subject object verb). У зв'язку з цим алгоритм для української мови має передбачати роботу з різним порядком слів.

Виділення кореферентних об'єктів дозволяє знаходити зв'язки між реченнями та всередині них, виділяти інформацію із текстів, покращувати результати аналізу текстів в інших задачах, таких, як переклад з однієї мови іншою, оцінка когерентності текстів.

На початкових етапах досліджень для пошуку кореферентних об'єктів використовували алгоритми на основі правил, сформованих досвідченими лінгвістами вручну. Такі алгоритми створювалися для конкретної мови та мали враховувати багато особливостей для досягнення якісного аналізу.

Із часом для розв'язання задачі почали використовувати автоматизовані підходи, такі як нейронні мережі та дерева рішень. Вони не вимагають створення правил вручну, але для їх підготовки необхідні великі набори даних. Часто в автоматизованих алгоритмах використовують прості правила для формування початкових кластерів.

У статті запропоновано метод пошуку кореферентних об'єктів з використанням дерев рішень. Розроблено застосунок, що використовує векторні представлення слів ELMo та інші характеристики для автоматизованого формування дерева рішень. Для формування дерева та оцінки точності роботи алгоритму використано набір підготовлених текстів, що містить понад 360000 тисяч слів. Підібрано параметри формування дерева. Дерево рішень, створене для визначення, чи пара об'єктів є кореферентними, використано для формування кластерів кореферентних об'єктів. Використано спеціальні метрики для порівняння з результатами, отриманими іншими алгоритмами в українській мові.

### Засоби, використані для реалізації алгоритму пошуку кореферентних об'єктів

Застосунок використовує векторні представлення слів, отримані за допомогою бібліотеки ELMo[2]. Ця бібліотека дозволяє перетворювати слова на вектори, що відповідають їхньому семантичному, лексичному, синтаксичному значенню. На відміну від інших бібліотек, які дозволяють формувати векторні представлення слів, таких як Word2Vec[3], векторні представлення, сформовані в ELMo, враховують не тільки значення окремого слова, а й значення навколишніх слів. Це дозволяє краще знаходити зв'язки між окремими словами та реченнями. ELMo використовує нейронні мережі для отримання векторних представлень слів та потребує навчання. Використана версія адаптована для української мови.

Використовується бібліотека Scikit-learn[4]. Ця бібліотека включає багато засобів для розв'язання задач регресії, кластеризації, класифікації. Зокрема, містить оптимізовану реалізацію дерев рішень із можливістю налаштування параметрів побудови дерева.

Однією з головних переваг дерев рішень над іншими алгоритмами є можливість візуалізації побудованого дерева. Для цього використано бібліотеку Graphviz[5].

Для створення дерева рішень використано підготовлений корпус україномовних текстів, що містить понад 360000 слів (> 2500 текстів). Розмітка кореферентних об'єктів у ньому проведена вручну, а для отримання додаткової інформації (рід, число лематизована (початкова) версія слова) використано бібліотеку UDpipe[6], що використовує нейронні мережі та навчена на україномовному корпусі текстів.

### Формат даних для представлення кореферентних об'єктів та їх аналізу

Для аналізу текстів за допомогою дерев рішень, їх потрібно подати у правильному вигляді. У роботі використані такі характеристики для опису кореферентних об'єктів (розглядалися у статті [7]):

- Косинусна схожість векторів семантичного представлення об'єктів, що розглядаються. Для цього використовувалася попередньо навчена модель ELMo. Якщо кількість слів, що входить в об'єкт > 1, використовується середнє арифметичне векторів слів.

- Кількість слів між обраними об'єктами.
- Кількість об'єктів між обраними потенційно кореферентними об'єктами.
- Булеве значення, правдиве, якщо перший об'єкт займенник.
- Булеве значення, правдиве, якщо другий об'єкт займенник.
- Булеве значення, правдиве, якщо лематизовані версії (початкові форми слів) об'єктів збігаються. В алгоритмі співпадіння лематизованих версій визначено як співпадіння хоча б одного слова в обох об'єктах, що розглядаються.

- Булеве значення, правдиве, якщо в обох об'єктах однакове число (однина чи множина).
- Булеве значення, правдиве, якщо в обох об'єктах однаковий рід.
- Булеве значення, правдиве, якщо обидва об'єкти власні назви.

На вхід алгоритму подається текст, що складається із окремих слів, пунктуації, та додаткової інформації, підготовленої за допомогою бібліотеки UDpipe. Сюди входить рід, число, лематизовані версії слова, частина мови. Також, для слів, що входять в кореферентні групи вказується ідентифікатор. Це дозволяє віднести конкретне слово або словосполучення до кореферентної групи (підготовлений вручну).

Для роботи алгоритму для кожного тексту, що розглядається, формуються список (Python list), що вміщуєш індекси слів, що входять в словосполучення, які є потенційно кореферентними об'єктами, до їх об'єднання в кореферентні кластери, але у форматі, що полегшує подальше їх об'єднання (1),  $w$  – окреме слово,  $[w_{i,1}, w_{i,2} \dots]$  - словосполучення.

$$ObjectsList = [\dots [[w_{i,1}, w_{i,2} \dots]], [[w_{i+1,1}, w_{i+1,2} \dots]] \dots] \quad (1)$$

Також створюються списки із правильно сформованими кластерами кореферентних об'єктів. Ці списки необхідні на етапі порівняння кластерів, отриманих у результаті передбачень дерева рішень та вірних кластерів (2),  $C_1$ ,  $[[w_{i,1}, w_{i,2}], [w_{j,1}, w_{j,2}]]$  – окремий кластер.

$$CorrectObjectsList = [C_1, C_2, \dots, [[w_{i,1}, w_{i,2} \dots], [w_{j,1}, w_{j,2} \dots] \dots], [[w_{i+1,1}, w_{i+1,2} \dots] \dots] \dots] \quad (2)$$

Оскільки задача кластеризації зведена в алгоритмі до задачі класифікації, для створення дерев рішень та передбачень з їхньою допомогою, необхідні списки з параметрами кожної пари потенційно кореферентних об'єктів (3), де *cosSim* – косинусна схожість об'єктів, *nWBtw* – кількість слів між об'єктами, що розглядаються, *nObjBtw* – кількість об'єктів між об'єктами, що розглядаються, *len1* – довжина (кількість слів) першого об'єкту, *len2* – довжина другого об'єкту, *1pron* – чи перший об'єкт – займенник, *2pron* – чи другий об'єкт – займенник, *1prp* – чи перший об'єкт – власна назва, *2prp* – чи другий об'єкт – власна назва, *lemS* – чи лематизовані версії об'єктів збігаються, *gendS* – чи об'єкти мають однаковий рід, *numS* – чи об'єкти мають однакове число.

$$X = [\cosSim_1, nWBtw_1, nObjBtw_1, len1_1, len2_1, 1pron_1, 2pron_1, 1prp_1, 2prp_1, lemS_1, gendS_1, numS_1, \dots] \quad (3)$$

Також необхідні мітки, що вказують, чи є пара об'єктів, що розглядаються, кореферентними (4).

$$Y = [y_1, y_2, y_3, \dots] \quad (4)$$

Після створення дерева рішень під час його використання для передбачень із списку (1) утворюється список (5), що містить перелік передбачених кластерів.

$$PredictedObjectsList = [C_1, C_2, \dots, [[w_{i,1}, w_{i,2} \dots], [w_{j,1}, w_{j,2} \dots] \dots], [[w_{i+1,1}, w_{i+1,2} \dots] \dots] \dots] \quad (5)$$

Отримані списки з характеристиками груп кореферентних об'єктів (3) а також їхнього маркування (4), що містять 2400000 зразків кореферентних та некореферентних об'єктів, діляться на дві частини – перша (1500 текстів, ~ 60%) використовується для формування дерева рішень, друга (1015 текстів, ~ 40%) – для перевірки результативності алгоритму (аналіз отриманих результатів).

## Алгоритм пошуку кореферентних об'єктів з використанням дерев рішень

**Формування дерева рішень:** Дерево рішень реалізоване з використанням бібліотеки *sklearn*, яка має клас дерева рішень *sklearn.tree.DecisionTreeClassifier*. При створенні екземпляру класу визначаються параметри дерева рішень – критерії розділення на піддерева під час формування, максимальна глибина дерева, мінімальна кількість елементів для розділення, мінімальна кількість елементів в одному листку дерева, ваги для класів, що передбачаються, тощо. За замовчуванням параметри дерева вказані для отримання безпомилкової конфігурації дерева на вибірці, що використовується для його формування. Такий підхід призводить до надмірної адаптації дерева до даних, що використовуються при його формуванні та знижує точність роботи на наборах, які раніше не аналізувалися алгоритмом. Також, для великих об'ємів даних внаслідок використання такої конфігурації створюється надміру велике дерево, формування якого забирає багато часу.

Отож, постає необхідність обмежити розміри дерева для досягнення вищих результатів на даних, що не використовувалися для формування дерева. Для цього застосовано параметр *min\_impurity\_decrease*, що дозволяє визначити мінімально достатнє значення зменшення неоднорідності в наступних піддеревих при розділенні. На відмінну від інших способів обмеження розмірів дерева, таких як обмеження глибини або обмеження на кількість елементів у листях, цей показник дозволяє рівномірніше обмежувати розміри дерева.

Для формування дерева підготовлені дані пар потенційно кореферентних об'єктів (3) із маркуванням, чи є вони кореферентними (4) подаються в функцію *fit*. На виході отримується дерево, здатне аналізувати пари кореферентних об'єктів. Таким чином задачу кластеризації кореферентних об'єктів зведено до задачі класифікації деревом рішень.

Частина отриманого дерева показана на Рис. 1. Для ілюстрації глибина дерева та кількість текстів, що аналізуються, штучно обмежена, щоб мати змогу помістити отримане дерево на екрані. Як видно з рисунка, розбиття на кореферентні та некореферентні об'єкти починається із характеристики, що дозволяє найкраще розділити поточну групу об'єктів – співпадіння лематизованих версій об'єктів. Далі, завдяки використанню дерев рішень, в піддереві дерева (Рис. 2) можна прослідкувати наступну логіку – якщо лематизовані версії об'єктів співпадають, довжина першого та другого об'єкту – 1 та перший об'єкт є власною назвою, тоді з високою імовірністю пара об'єктів є кореферентними.

**Використання дерева рішень для пошуку кореферентних об'єктів:** після формування дерева рішень створене дерево використовується для отримання кластерів кореферентних об'єктів: відбувається перебір пар об'єктів – кандидатів на кореферентність (1). Для кожної пари об'єктів визначаються параметри (3), необхідні для класифікації об'єктів деревом рішень. Якщо пара класифікована як кореферентна, відбувається злиття об'єктів. Після утворення кластерів, що містять декілька об'єктів, їх злиття відбувається, якщо хоча б одна пара об'єктів із першого та другого кластеру розпізнана кореферентною. Під час експериментальних досліджень виявлено, що при використанні декількох циклів проходів, поки можливе злиття кластерів, результати на метриках (наступний пункт) виявлялися на 2-5% вищими, ніж без використання циклічних проходів. Тому в фінальній версії алгоритму використовується декілька проходів, до поки в циклі відбувається хоча б одне злиття. В результаті роботи алгоритму утворюється список кластерів, що містить кореферентні об'єкти всередині спільних кластерів(5).

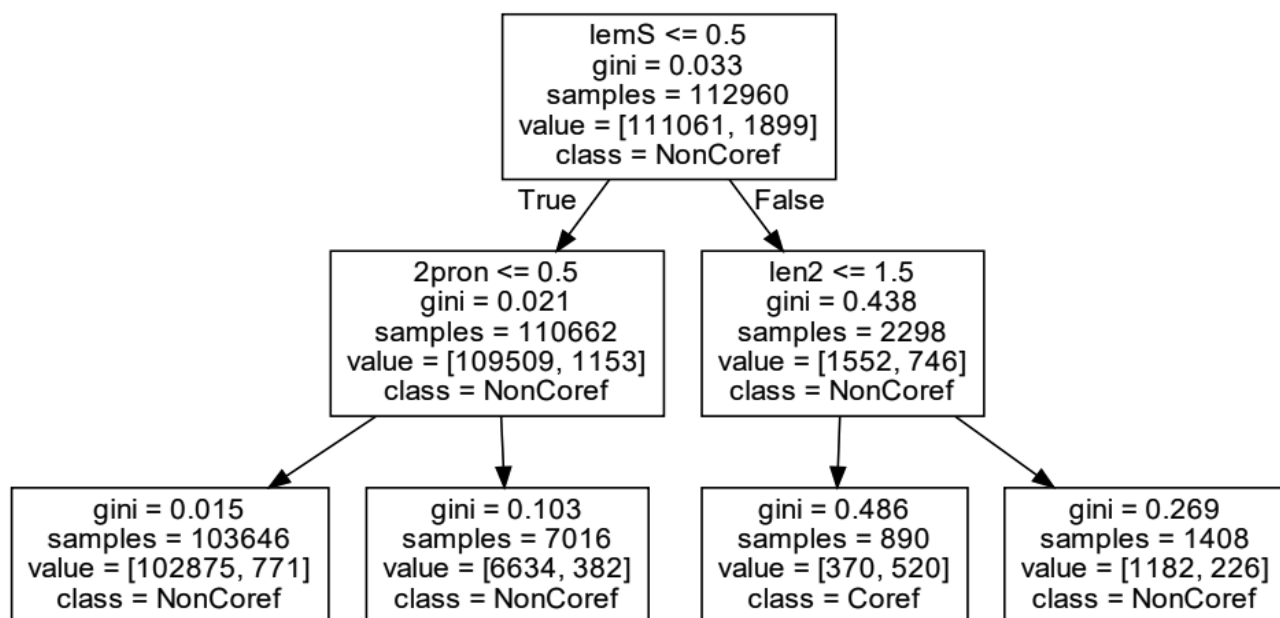


Рис. 1. Дерево рішень

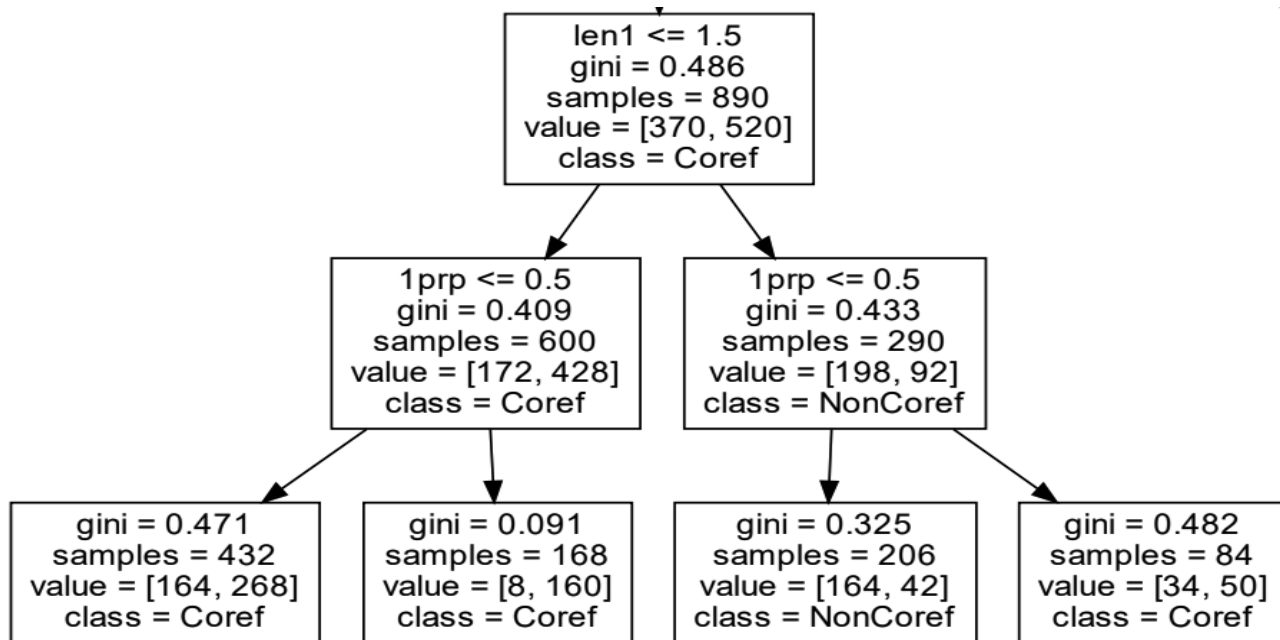


Рис. 2. Піддерево дерева рішень

### Аналіз отриманих результатів

Для оцінки результатів, отриманих внаслідок використання алгоритму пошуку кореферентних об’єктів, використовуються метрики  $b^3$  [8] і MUC [9]. Ці метрики дозволяють порівнювати групи кластерів – із правильним упорядкуванням кореферентних об’єктів (2) та із передбачуваним упорядкуванням (5), чисельно відображаючи різницю між кластерами. Для кожної метрики обчислюється влучність – precision (відношення вірно обраних алгоритмом об’єктів до усіх обраних об’єктів), повнота – recall (відношення вірно обраних алгоритмом об’єктів до усіх об’єктів, що належать до даного кластеру), та F1 міра (середнє гармонійне повноти та влучності).

Метрика  $b^3$  використовується в широкому колі задач кластеризації. Метрика  $b^3$  розглядає окремі елементи в списку передбачених кластерів, для яких розраховується інтегральний показник. Влучність для метрики  $b^3$  визначається як середнє арифметичне влучностей для кожного елемента:

$$P = \frac{1}{N} \sum_N \frac{Z_n}{T_n} \quad (6)$$

де  $n$  – номер обраного елемента,  $N$  – кількість елементів у списку,  $Z_n$  – кількість елементів, що належать до тої ж кореферентної групи, що й обраний елемент (включаючи обраний елемент), та входять у передбачуваний кластер,  $T_n$  – загальна кількість елементів у передбаченому кластері.

Повнота для метрики  $b^3$  визначається як середнє арифметичне повноти для кожного елемента:

$$R = \frac{1}{N} \sum_N \frac{Z_n}{M_n} \quad (7)$$

де  $n$  – номер обраного елемента,  $N$  – кількість елементів у списку,  $Z_n$  – кількість елементів, що належать до тої ж кореферентної групи, що й обраний елемент (включаючи обраний елемент) та входять в передбачуваний кластер,  $M_n$  – загальна кількість елементів в тій же групі, що й обраний елемент.

Метрика MUC спеціально розроблена для оцінки ефективності роботи алгоритмів, що розв’язують задач пошуку кореферентних об’єктів. У метриці MUC розглядається весь список кластерів, для якого розраховуються показники. Для MUC повнота визначається формулою:

$$R = \frac{\sum(|S_i| - |P(S_i)|)}{\sum |S_i|} \quad (8)$$

де  $|S_i|$  – кількість елементів в істинному кореферентному кластері,  $|P(S_i)|$  – кількість підгруп, на які істинний кореферентний кластер ділять передбачувані кластери. Влучність визначається як:

$$P = \frac{\sum(|S_i| - |P'(S_i)|)}{\sum |S_i|} \quad (9)$$

де  $|S_i|$  – кількість елементів у передбачуваному кореферентному кластері,  $|P'(S_i)|$  – кількість підгруп, на які передбачуваний кореферентний кластер ділять істинні кластери.

Для кожної метрики обчислюється F1 міра, що визначається формулою:

$$F_1 = \frac{2 \cdot R \cdot P}{R + P} \quad (10)$$

де  $R$  – повнота,  $P$  – влучність.

Оцінка результатів роботи отриманого дерева рішень виконується на частині корпусу, що не використовувалася під час його формування (1015 текстів). Під час проведення експериментальних досліджень визначено оптимальне значення параметру `min_impurity_decrease`, за якого досягнуто найвищих результатів на метриках  $b^3$  і MUC. Результати порівняння роботи з іншими алгоритмами, що використовувалися для аналізу країномовних текстів [1, 7], наведені в табл. 1.

Таблиця 1. Порівняння ефективності алгоритмів на метриках  $b^3$  та MUC

Модель	Метрика	MUC	$B^3$
CNN	Влучність	24.23	97.88
	Повнота	12.45	84.99
	F1	16.44	92.11
BiLSTM (1 прохід)	Влучність	56.91	95.94
	Повнота	30.20	88.76
	F1	29.46	92.21
BiLSTM (декілька проходів)	Влучність	56.36	93.13
	Повнота	39.68	90.43
	F1	45.88	91.76
RoBERTa	Влучність	27.39	91.22
	Повнота	13.10	89.65
	F1	17.72	90.43
Дерево рішень	Влучність	73.46	98.15
	Повнота	29.08	88.14
	F1	41.67	92.87

Для метрики  $b^3$  дерево рішень показує найвищі результати порівняно з іншими підходами. На метриці MUC підхід з використанням дерев рішень показує значно вищі результати, ніж досягнуті з використанням згорткової нейронної мережі (CNN) [7] та моделі природної мови RoBERTa [1], на рівні двозв’язної нейронної мережі з довгою та короткочасною пам’яттю (BiLSTM) [7]. Варто зазначити, що влучність алгоритму для даного дерева рішень на метриці MUC та  $b^3$  найвища порівняно з іншими підходами, водночас повнота перебуває на рівні варіанту BiLSTM з одним проходом. При цьому параметри формування дерева рішень дозволяють підвищити його влучність за рахунок зниження повноти або навпаки, тим самим адаптуючись до випадків, для яких важливіша точність для знайдених кореферентних об’єктів (влучність) чи знаходження більшої частини кореферентних об’єктів (повнота).

### Висновки

У процесі виконання роботи розглянуто задачу пошуку кореферентних об'єктів в українській мові. Створено застосунок, що використовує дерева рішень для пошуку кореферентних об'єктів: підготовлено дані для аналізу, побудовано дерево рішень, модифіковано його параметри для досягнення вищих результатів, оцінено продуктивність застосунку на метриках  $b^3$  і MUC. Дерева рішень завдяки можливості графічного представлення дозволяють полегшити аналіз отриманих результатів та пояснити, як саме отримується той чи інший результат, що відрізняє їх від інших популярних методів аналізу даних, зокрема, нейронних мереж.

Отримані результати показують порівняно високу ефективність алгоритму – досягнуто найвищі показники на метриці  $b^3$  та порівняно високі на метриці BiLSTM, із найвищими показниками влучності, що вказує на доцільність використання дерев рішень для пошуку кореферентних об'єктів в українськомовних текстах.

### Література

1. С.Д. Погорілий, П.В. Білецький. Використання графічного процесора для прискорення пошуку кореферентних об'єктів з використанням моделі RoBERTa. Наукові праці ДонНТУ, Серія: "Інформатика, кібернетика та обчислювальна техніка", 2022. № 2. С. 4-9.
2. M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, № 1. С. 2227–2237.
3. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed Representations of Words and Phrases and their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, № 2, С. 3111–3119.
4. Бібліотека Scikit-learn. <https://scikit-learn.org/>
5. Бібліотека Graphviz. <https://graphviz.org/>
6. Бібліотека UDpipe. <https://lindat.mff.cuni.cz/services/udpipe/>
7. S. Telenyk, S. Pogorilyy, A. Kramov. The complex method of coreferent clusters detection based on a BiLSTM neural network, Knowledge Based Systems. 2021. С. 205-210.
8. A. Bagga, B. Baldwin, Algorithms for Scoring Coreference Chains, The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, 1998, С. 563-566.
9. M. Vilain, J. Burger, J. Aberdeen, D. Connolly, L. Hirschman, A Model-Theoretic Coreference Scoring Scheme, Proceedings of the 6th Conference on Message Understanding (MUC), 1995.

### References

1. POGORILYY S. & BILETSKYI P. (2022) Usage of a graphics processor to accelerate coreference resolution while using the RoBERTa model. Scientific works of DonNTU, Series: "Informatics, cybernetics and computer technology". (2) P. 4-9.
2. PETERS M., NEUMANN M., IYER M., GARDNER M., CLARK C., LEE K., ZETTLEMOYER L.. (2018) Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (1). P. 2227–2237.
3. MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G., DEAN J.. (2013) Distributed Representations of Words and Phrases and their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems, (2). P. 3111–3119.
4. Scikit-learn library. <https://scikit-learn.org/>
5. Graphviz library. <https://graphviz.org/>
6. UDpipe library. <https://lindat.mff.cuni.cz/services/udpipe/>
7. TELENYK S., POGORILYY S., KRAMOV A. (2021) The complex method of coreferent clusters detection based on a BiLSTM neural network, Knowledge Based Systems. P. 205-210.
8. BAGGA A., BALDWIN B., (1998) Algorithms for Scoring Coreference Chains, The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, P. 563-566.
9. VILAIN M., BURGER J., ABERDEEN J., CONNOLLY D., HIRSCHMAN L. (1995) A Model-Theoretic Coreference Scoring Scheme, Proceedings of the 6th Conference on Message Understanding (MUC).

Одержано 17.08.2022

### Про авторів:

*Погорілий Сергій Дем'янович*

доктор технічних наук,

професор,

Кількість публікацій в українських виданнях – 109.

Кількість зарубіжних публікацій – 67.

<https://orcid.org/0000-0002-6497-5056>.

*Білецький Павло Володимирович,*

аспірант,

Кількість публікацій в українських виданнях – 3.

<https://orcid.org/0000-0002-4248-4766>.

**Місце роботи авторів:**

Факультет радіофізики, електроніки та комп'ютерних систем,  
Київський національний університет імені Тараса Шевченка,  
03187, м. Київ, проспект Академіка Глушкова, 4г  
Тел.: (38)(044) 521-05-32  
E-mail: [rex@knu.ua](mailto:rex@knu.ua)

**Прізвища та ініціали авторів і назва доповіді англійською мовою:**

Pogorilyy S. D., Biletskyi P. V.  
Coreference resolution algorithm for Ukrainian-language texts using decision trees

**Прізвища та ініціали авторів і назва доповіді українською мовою:**

Погорілий С. Д., Білецький П. В.  
Алгоритм пошуку кореферентних об'єктів в україномовних текстах  
з використанням дерев рішень

**Контакти для редактора:**

Погорілий Сергій Дем'янович, доктор технічних наук, професор,  
кафедра Комп'ютерної інженерії факультету радіофізики,  
електроніки та комп'ютерних систем,  
Київський національний університет імені Тараса Шевченка,  
e-mail: [sdp77@i.ua](mailto:sdp77@i.ua).

Білецький Павло Володимирович, аспірант,  
кафедра Комп'ютерної інженерії факультету радіофізики,  
електроніки та комп'ютерних систем,  
Київський національний університет імені Тараса Шевченка,  
e-mail: [1234bprv@i.ua](mailto:1234bprv@i.ua), тел.: (38)(050) 735-59-35.