МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ТЕЛЕЦЬКА АЛІНА ОЛЕКСАНДРІВНА

Допускається до захисту:
Завідувач кафедри загального
та прикладного мовознавства
і слов'янської філології,
доктор філологічних наук, доцент
_____ Г.В. Ситар
«_____» _____ 20___ р.

ТИПОЛОГІЯ МОРФОЛОГІЧНИХ І СИНТАКСИЧНИХ МОДЕЛЕЙ: ЛІНГВАЛЬНИЙ І ФОРМАЛЬНИЙ ВИМІРИ (НА МАТЕРІАЛІ ТЕКСТІВ ПРОЕКТУ UKRAÏNER)

Спеціальність 035 Філологія

Кваліфікаційна робота

Науковий керівник:
А. П. Загнітко,
професор кафедри загального та
прикладного
мовознавства і слов'янської філології,
доктор філологічних наук, професор
_____
(підпис)

Оцінка: _____ / _____ /
_____
(бали за шкалою ЄКТС/за національною шкалою)
Голова ЕК: _____
(підпис)

Вінниця 2021

АНОТАЦІЯ

Телецька А. О. Типологія морфологічних і синтаксичних моделей: лінгвальний і формальний виміри (на матеріалі текстів проекту Ukraïner). Спеціальність 035.10 «Прикладна лінгвістика», Освітня програма «Прикладна лінгвістика», Донецький національний університет імені Василя Стуса, Вінниця, 2021.

У кваліфікаційній роботі досліджено морфологічні та синтаксичні моделі для української мови, що можуть допомогти в розробці автоматизованих комп'ютерних систем, а також проаналізовано засоби обробки природніх мов, напрацьовано алгоритми кваліфікації морфологічних і синтаксичних моделей. Обґрунтовано автоматичний синтаксичний аналіз зі допомою бібліотеки Spacy.

Ключові слова: моделі, лінгвістичні моделі, синтаксичний аналіз, природна обробка мови, машинне навчання, бібліотеки Python, Spacy.

63 с., рис. 26, джерел 46.

Teletska A. Typology of morphological and syntactic models: the lingval and formal survey (based on the texts of the project Ukraïner). Specialty 035.10 «Applied Linguistics», Programme «Applied Linguistics». Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

The final year project investigates morphological and syntactic models for the Ukrainian language that can help in the development of automated computer systems. It analyzes the technologies of Natural Language Processing, algorithms for qualification of morphological and syntactic models have been developed. Automatic parsing with the help of the Spacy library is substantiated

Keywords: models, linguistic models, parsing, natural language processing, machine learning, Python libraries, Spacy.

63 p., fig. 26., bibliography: 46 items.

ЗМІСТ

ВСТУП

Автоматичне розпізнавання текстів природньої мови є одним із найпоширеніших завдань сучасних науковців. Саме через системи моделювання текст перетворюється на зрозумілу мову для комп'ютерних технологій аналізу та синтезу природніх мов. Сьогодні для української мови практично немає у відкритому доступі моделей, що пройшли навчання й за допомогою яких можна проводити глибокий автоматичний аналіз текстів. Тому дослідження полягає у висвітленні засобів навчання автоматизованих моделей й спостереженні аналізу текстів української мови через доступні моделі інших мов.

Актуальність теми дослідження зумовлена потребою розвитку автоматизації обробки текстів. У такий спосіб з'являється необхідність проаналізувати теоретичні надбання про моделювання граматичних структур для української мови та висвітлити технології й засоби для автоматизації цих моделей на прикладі систем для інших мов.

Вагомий внесок у розвиток автоматичного синтаксичного аналізу текстів природньої мови зробили Ноам Хомський (синтаксичні структури), Джон Бекус (синтаксис формальних мов), Джоакім Нівре (граматика залежностей), Даніель Юрафський, Джеймс Мартін (Обробка природньої мови). Серед вітчизняних науковців над темою автоматичного синтаксичного аналізу досліджували й дотепер працюють: В.Г. Волошин, Т.А. Грязнухіна, Н.П. Дарчук, І.П. Білецька І. П.

Об'єкт дослідження – морфологічні та синтаксичні моделі української мови.

Предмет дослідження – типологія морфологічних та синтаксичних моделей з лінгвістичного погляду та їх формалізація.

Мета роботи полягає в дослідженні морфологічних та синтаксичних моделей для української мови, що можуть стати основою розроблення автоматизованих комп'ютерних систем.

Досягнення мети передбачає виконання низки **завдань**:

– скласифікувати типи морфологічних та синтаксичних моделей;

– дослідити алгоритми автоматичної синтаксичної обробки текстів;

– перевірити роботу засобів обробки текстів української мови з допомогою доступних навчених моделей;

– з'ясувати, які технології машинного навчання можна застосувати для навчання власних автоматизованих моделей, що виконуватимуть синтаксичний аналіз.

Методи дослідження. Під час дослідження в роботі були використані такі методи: аналіз, систематизація даних про автоматичний синтаксичний аналіз текстів, класифікування матеріалу, синтез і узагальнення інформації про моделювання мовної системи на граматичному рівні.

Наукова новизна дослідження полягає у напрацюванні синтаксичних моделей для автоматичного синтаксичного аналізу в бібліотеці Spacy, а також у встановленні застосовності теоретичного матеріалу про граматичну структуру для дослідження її формалізації й адаптації для комп'ютерних систем.

Практичне значення отриманих результатів про автоматичний синтаксичний аналіз текстів та моделі граматичних структур української мови можуть бути корисним у майбутній розробці власної автоматизованої моделі**.**

Апробація результатів дослідження підтверджена публікацією статті в міжнародному видання: Телецька А. О. Засоби оброблення природнього мовлення / А. О. Телецька, А. П. Загнітко // Modern directions of scientific research development / А. О. Телецька, А. П. Загнітко. – Chicago, USA, 2021. – (BoScience Publisher). – (The 5th International scientific and practical conference). – С. 739–746.

Структура та обсяг роботи. Робота складається із вступу, трьох розділів, висновків, списку літератури (72 позиції, з них: використана література (46), джерела ілюстративного матеріалу (26)). Загальний обсяг роботи становить 76 сторінки, з них основного тексту – 63 сторінки.

РОЗДІЛ 1.
МОРФОЛОГІЧНЕ ТА СИНТАКСИЧНЕ МОДЕЛЮВАННЯ

2.2. Модель та моделювання

У сучасному світі ми можемо спостерігати швидкий розвиток технологій. Наука досягла значних успіхів у дослідженнях як на молекулярному рівні, так і у світі космічних тіл. Базуючись на закономірностях природних явищ людина змогла й далі продовжує наслідувати у власних винаходах різні системи. Багато новітніх розробок пов'язані із роботою людського мозку – це відома всім інженерна система штучного інтелекту. Цікавою розробкою насьогодні стали 3D-принтери. Вони можуть значно вдосконалити виробничий процес шляхом 3D-друку від виробництва предметів побуту до лікарських засобів та замінників для деталей різних машин. Щоб досягти успішного керування власним бізнесом підприємцю чи всій компанії загалом потрібно створити логічний план, який чітко відобразить стратегії ведення цього бізнесу. Усі ці речі пов'язані з моделюванням – дослідженням явища чи об'єкта через спостереження та експерименти з його моделлю [1, с. 33].

За І. М. Кульчицьким, модель – це система-репрезентант, аналіз якої слугує способом отримати інформацію про іншу систему [7, с. 280]. Іншими словами, модель – це абстракція дійсності або зображення реального об'єкта чи ситуації, тобто представлення спрощеного варіанту певного явища.

Також однією з найпоширеніших концепцій моделі є така, що це абстракція від реальної проблеми, ключових змінних та взаємозв'язків. Вони абстраговані для спрощення самої проблеми. Моделювання дає змогу користувачеві краще зрозуміти проблему і представляє засіб для маніпулювання ситуацією з метою аналізу результатів різних вхідних джерел, під час яких відбуваються зміни в наборі припущень [29].

1.1.1. Класифікації моделей

Деякі моделі є копіями фізичних властивостей (кольору, форми, ваги і т. ін.) об'єкта, який вони представляють. Інші – є фізичними моделями, але мають

не такий зовнішній вигляд, як об’єкт їх подання. Третій тип моделі стосується символів, числових відношень та виразів. Тому виділяють такі основні категорії моделей: фізичні, схематичні, словесні та математичні.

Фізичні моделі – це ті, що схожі на готовий об’єкт, який вони репрезентують. Традиційно такими точними або надзвичайно схожими копіями є моделі літаків, автомобілів і кораблів. Вони виглядають точно так само, але в значно менших масштабах. Перевагою тут є відповідність моделей у зовнішності, тобто користувач моделі може точно сказати, як буде виглядати запропонований об’єкт.

Схематичні моделі більш абстрактні, ніж фізичні моделі. Хоча вони і мають певну візуальну відповідність реальності, проте набагато менше схожі з фізичною реальністю, яку вони представляють. Графіки та діаграми – це схематичні моделі, що забезпечують графічне зображення математичних взаємозв’язків. Побудова лінії на графіку вказує на математичну лінійну залежність між двома змінними. Кругові діаграми та гістограми можуть моделювати якусь реальну ситуацію, але насправді нічим не будуть нагадувати її фізично.

Схеми та креслення також є версіями схематичних моделей. Це зображальні уявлення про концептуальні зв’язки (блок-схема, що описує комп’ютерну програму).

Словесні моделі, або **вербальні** використовують слова, щоб зобразити якийсь об’єкт або ситуацію, яка існує або могла б існувати насправді. Словесні моделі часто надають сценарій, необхідний, щоб вказати на наявність проблеми, і надають всю відповідну та необхідну інформацію для вирішення проблеми, надання рекомендацій або, принаймні, визначення можливих альтернатив. Словесні моделі часто перетворюються на математичні моделі, щоб можна було знайти оптимальне або, принаймні, функційне вирішення, використовуючи деякі математичні методи.

Математичні моделі взагалі не схожі на свої аналоги в реальному житті. Вони будуються з використанням чисел і символів, які перетворюють на

функції, рівняння та формули. Вони також можуть бути використані для побудови набагато складніших моделей, таких як матриці або моделі лінійного програмування. Потім користувач може обрахувати математичну модель (віднайти оптимальну відповідь), використовуючи прості прийоми, такі як множення та додавання, або більш складні прийоми, застосовуючи матричну алгебру [29].

1.1.2. Переваги та недоліки використання моделі

Мета використання моделювання – зображення належним чином певного існуючого явища. Після правильної розробки можна багато дізнатися про реальний аналог, керуючи змінними моделі та спостерігаючи за результатами.

Моделі дають змогу користувачеві усувати неважливі деталі, щоб користувач міг зосередитися на відповідних змінних рішення, які присутні в ситуації. Це збільшує можливість повного розуміння проблеми та її вирішення.

Вільям Дж. Стівенсон перераховує дев'ять переваг моделей:

1. Моделі зазвичай прості у використанні та дешевші, ніж у реальній ситуації.
2. Моделі вимагають від користувачів упорядкування, а іноді і кількісної оцінки інформації, і в процесі часто вказують сфери, де потрібна додаткова інформація.
3. Моделі забезпечують системний підхід до вирішення проблем.
4. Моделі покращують розуміння проблеми.
5. Моделі дають змогу менеджерам аналізувати питання «що якщо».
6. Моделі вимагають від користувачів досить конкретних цілей.
7. Моделі служать послідовним інструментом оцінки.
8. Моделі дають змогу користувачам використати силу математики для вирішення проблеми.
9. Моделі забезпечують стандартизований формат для аналізу проблеми [29].

Моделі не завжди дають очікуваний результат. Їх використання має певні обмеження, що часто в результаті може привести до похибки. Модель – це

спрощення поставленої проблеми чи завдання, і потрібно пам'ятати, що не завжди можна зробити повний та точний аналіз.

1.2. Лінгвістичні моделі

За Володимиром Волошиним, мова – це явище суспільно-історичне і служить засобом спілкування в людському суспільстві. Наш мозок через різні складні процеси здатен згенерувати думку, яку ми передаємо під час мовлення [3, с. 23].

Людське мовлення не є чимось хаотичним та випадковим. Це відтворювана система, що має власну закономірність, керується внутрішніми та зовнішніми зв'язками й відношеннями, які є особливими для різних мов. Письмо – зображувана форма мовлення, через яку можна в знаках та символах передати інформацію та емоційний стан мовця. Оскільки структура мовлення відстежується, її почали детальніше досліджувати на всіх рівнях мови. Моделювання як найдоступніший метод дослідження для опису будови та функціювання мови набув широкого вжитку.

У лінгвістичній літературі термін «модель» вперше було вжито 1944 року американцем Зеллігом Гаррісом, характеризуючи різновиди методологічних прийомів двох лінгвістів Хьюмена й Сепіра. У 1951 році він ще використав цей термін для позначення результатів описової методології Сепіра. Однак у більш конкретному значенні термін «модель» і саме в застосуванні до граматики був використаний у 1954 році Чарльзом Хокеттом, а також у 1956 р. Ноамом Хомським. Ці дослідники вжили термін «модель» у значенні узагальненої й формалізованої структури або процесу тих чи тих мовних якищ. В 1957 р. А. Еттінджер вже говорить про моделі не тільки як про відтворення дійсності, але і як про зворотню дію на цю дійсність. Погляди Еттінджера перетинаються з поглядами тих, хто в цей час уже намагався зіставити мову машин і мову людини. В. Інгве поширив розуміння моделі з конкретних мов на механзм мови взагалі [3, с. 14].

Для того, щоб описати мовленнєву діяльність, виділяють два типи моделей – описову й відтворюючу. Моделі описового (пояснювального) типу

застосовуються, щоб проаналізувати безпосередньо мовлення людини. Моделі відтворюючого типу – це породження мовного продукту, близького до того, який створила би людина. Наприклад, переклад людиною з однієї природної мови на іншу. Також моделі відтворюючого типу стосуються й машинного перекладу, результатом дослідження якої є розробка алгоритмів та програм.

Мовленнєва комунікація між складається з мотивації мовця та комунікативної інтенції. Сприйняття іншою особою цього повідомлення відбувається через обробку інформації в центральній і периферійній нервовій системах акустичного мовленнєвого сигналу [3 с. 25]. На цьому рівні також виділяють дослідницькі моделі, а саме нейролінгвістичні та психолінгвістичні.

Нейролінгвістичні моделі досліджують зв'язок між будь-якою зовнішньою мовною діяльністю людини та відповідною електричною та гуморальною діяльністю нервів у їх мозку. Було б цікаво детально дослідити, яка частина мозку активується, коли людина готується й виголошує висловлювання або намагається зрозуміти висловлювання, щойно почуте. На жаль, проблема виявлення способу мислення людей під час розмови чи розуміння надзвичайно складна. Дійсно, унікальним об'єктивним способом дослідника розкрити способи людського мислення нейрофізіологічними методами є синхронне дослідження електричної та гуморальної діяльності в багатьох місцях людського мозку.

Для обчислювальної лінгвістики існує проблема, як вибрати вхідні та вихідні сигнали для моделювання мозкової діяльності, залишається незрозумілою. Якщо ми запропонуємо деякі конкретні внутрішні уявлення для цих цілей, то формальні нейрони будуть лише моделювати наші винаходи, а не реальні мовні процеси. З цієї причини без революційних змін у методах дослідження та нових підходів до обробки даних, що підлягають спостереженню, нейролінгвістичні моделі розуміння природної мови навряд чи дадуть хороші результати найближчим часом. В даний час можна ефективно спостерігати лише дуже грубі особливості мозкової діяльності, наприклад, які

ділянки мозку виявляють нейрональну активність, пов’язану з пам’яттю людини, загальну здатність міркувати тощо.

Ще однією моделлю складною моделлю є психолінгвістична модель. Психолінгвістика – це наука, що досліджує мовленнєву діяльність людини, включаючи сприйняття та формування висловлювань, за допомогою психологічних методів. Створивши свої гіпотези та моделі, психолінгвістика перевіряє їх за допомогою психологічних експериментів. Отже, психолінгвістика схожа на лінгвістику за своїми об’єктами дослідження, а також є аналогічною психології за своїми методами.

На основі експериментів дослідники висувають гіпотезу про те, що різні психологічні типи людей демонструють конкретні типи асоціативних рішень. У такий спосіб вони можуть дати правдиву ідентифікацію досліджуваної особистості на грунті подібних тестів. На підставі експериментальної статистики також проводяться дослідження, де висувається гіпотеза про те, як людина може розуміти невідомі їх конструкції.

Психолінгвістика намагається теж описати викладання рідної та іноземної мови, соціальний вплив мовлення на людину тощо. Таким чином, психолінгвістика має на меті пояснити деякі психологічні особливості людей щодо їх мовленнєвої поведінки. Психологічні особливості, у свою чергу, тісно пов’язані із соціальною поведінкою людини.

Тому психолінгвістика зазвичай не має власних лінгвістичних уявлень. Вона приймає їх із різних галузей мовознавства, а потім використовує для своїх цілей без особливого критичного огляду та зворотного зв’язку [16, с. 129-133].

1.2.1. Функційні моделі мови

Проблема кібернетичного моделювання природної мови є складнішою, ніж в інших випадках, оскільки існує дві такі скриньки, аналізувальна та синтезувальна, що працюють у протилежних напрямках. Аналізувальний блок обробляє висловлювання, а синтезувальний блок виробляє реакції на них.

Дослідник спостерігає за входом аналізувального блоку та за висновком синтезувального блоку й намагається реконструювати внутрішню структуру

кожного блоку окремо. На жаль, вихід аналізатора не використовується безпосередньо як вхід синтезатора. Між ними є блок міркувань, і його поведінка не описується лінгвістичними термінами, так що його не так легко розпізнати.

Основним методом лінгвістики є побудова моделі природної мови, заснованої на спостережуваних вхідних та вихідних текстах, а також на інтуїції або самоаналізі лінгвіста. Лінгвісти аналізують власну інтуїцію, висувають гіпотези, будують моделі та перевіряють їх на додатковому мовному матеріалі. У теоретичній лінгвістиці нові підходи можуть бути перевірені інтуїціями інших лінгвістів, тоді як в обчислювальній лінгвістиці ці підходи також можуть бути перевірені за допомогою різних застосувань.

Таким чином, лінгвісти запропонували функційні моделі мови. Ці моделі призначені для надання правил перетворення вхідної лінгвістичної інформації у вихідну інформацію без будь-яких спроб безпосередньо відтворити внутрішні механізми мозкової діяльності. Жодних антропоморфних особливостей етапів обробки не шукають, а також прямої емуляції реакцій мозку не проводиться. Однак кінцеві результати всіх етапів обробки повинні бути якомога ближчими до результатів людського мозку.

На сьогодні функційні моделі виявились найуспішнішими лінгвістичними моделями, мабуть, тому, що вони базуються на реальних даних із мислимою структурою, легко доступними та доступними в необмеженій кількості, а саме на текстах та записаному мовленні.

1.2.2. Дослідження лінгвістичних моделей

Існують ще й інші моделі, що цікавлять лінгвістику. Їх називають дослідницькими моделями. При введенні вони приймають тексти природною мовою, можливо, заздалегідь підготовлені або відформатовані спеціальним чином. На виході вони створюють інші тексти зазвичай суворо відформатовані та представляють зміст словників, граматичних таблиць, правил чи чогось подібного, що використовується як частина функційних моделей.

Як приклад, ми можемо зібрати всі узгоджені пари, такі як «дієслово – іменник» або «іменник – прикметник», або всі прийменники, що трапляються у відкритому, тобто не підготовленому, тексті природною мовою. Як інший приклад ми можемо витягти з тексту словника ті предмети певної частини мови, які містять заздалегідь визначену комбінацію ознак.

Отже, дослідницькі моделі – це інструменти побудови функційних моделей. Вони імітують лінгвістів у своїх дослідженнях, тоді як функційні моделі імітують людей у мовленні, що виробляє та розуміє [16, с. 134].

1.2.3. Особливості сучасних моделей мови

Сучасні моделі мови мають декілька спільних рис, досить важливих для їхнього розуміння та використання. Однією з таких моделей є теорія «Смисл ↔ Текст». Іншою моделлю є та, що базується на формальній, генеративній теорії граматики (HPSG). Підхід Хомського в рамках західної лінгвістичної традиції включає різні інші моделі, відмінні від формальної, генеративної теорії граматики.

Основні загальні риси всіх цих моделей:

- Функційність моделі. Лінгвістичні моделі намагаються відтворити функції мови без безпосереднього відтворення особливостей діяльності мозку, який є рушієм людської мови.
- Протиставлення текстової / фонетичної форми мови її семантичному відображенню. Зовнішньою, що спостерігається, формою мовної діяльності є текст, тобто рядки фонетичних символів або букв, тоді як внутрішня, прихована форма тієї самої інформації є змістом цього тексту. Мова пов’язує дві ці форми однієї і тієї ж інформації.
- Узагальнювальний характер мови. Окремі висловлювання в межах виступу чи тексту розглядаються не як мова, а як зразки її функціонування. Мова є теоретичним узагальненням відкритих і, отже, нескінченного набору висловлювань. Узагальнення включає ознаки, типи, структури, рівні, правила тощо, які безпосередньо не можна спостерігати. Швидше ці теоретичні конструкції є плодами інтуїції лінгвіста і

підлягають неодноразовому тестуванню на нових висловлюваннях та інтуїції інших лінгвістів. Особливість узагальнення пов'язана з опозиційною компетенцією та виконанням у теорії Хомського та набагато раніше мовою опозиції проти мови у теорії Фердинанда де Соссюра.

- Динамічний характер моделі. Функційна модель не лише пропонує набір лінгвістичних понять, але також показує (за допомогою правил), як ці поняття використовуються при обробці висловлювань. Функційна модель – це система правил, достатньо сувора, щоб застосовуватись до будь-якого тексту людиною чи автоматом цілком формально, без втручання автора моделі чи когось іншого. Застосування правил до даного тексту або значення надає завжди однаковий результат. Будь-яка частина функційної моделі в принципі може бути виражена в суворій математичній формі і таким чином алгоритмізована. Якщо в даний час немає готового математичного інструмента, потрібно створити новий інструмент. Припущені властивості впізнаваності та алгоритмічності природної мови досить важливі для лінгвістичних моделей, спрямованих на комп'ютерну реалізацію.
- Негенеративний характер моделі. Інформація не виникає і не генерується в рамках моделі; вона лише набуває форми, що відповідає іншому мовному рівню. В оригінальних генеративних граматиках Хомського рядки символів, які можна інтерпретувати як висловлювання, що породжуються з початкового символу, який має абстрактний зміст речення.
- Незалежність моделі від напрямку трансформації. Опис мови не залежить від напрямку лінгвістичної обробки. Якщо обробка підпорядковується деяким правилам, ці правила потрібно надавати в рівноправній (тобто, зберігаючи сенс) двонаправленій формі, інакше вони повинні давати змогу зворотне використання в принципі.
- Незалежність алгоритмів від даних. Опис мовних структур слід розглядати окремо від алгоритмів, що використовують цей опис. Знання

мови не передбачає певного типу алгоритмів. Навпаки, у багатьох ситуаціях алгоритм, що реалізує деякі правила, може мати безліч варіантів. Наприклад, теорія «Смисл ↔ Текст» описує рівень тексту окремо від морфологічного й синтаксичного рівнів подання того самого висловлювання. Тим не менше, можна уявити алгоритм аналізу, який починає будувати відповідну частину синтаксичного подання так само, як формується морфологічне подання першого слова у висловлюванні. У випадках, коли лінгвістичні знання подаються в декларативній формі з максимально можливою послідовністю, алгоритми реалізації виявились досить універсальними, тобто однаково застосовними до кількох мов. Аналогічне розмежування між алгоритмами та даними з великим успіхом використовується у сучасних компіляторах мов програмування.

- Словники мови. Основна частина опису будь-якої мови передбачає слова цієї мови. Отже, словники, що містять описи окремих слів, вважаються основною частиною детального мовного опису. Лише дуже загальні властивості великих класів та підкласів лексем абстраговані від словників, щоб складати формальну граматику [16, с. 134-137].

1.2.4. Специфічні особливості моделі «Смисл-Текст»

Орієнтація на синтез. Внаслідок еквівалентності напрямків синтезу та аналізу синтез вважається основним та важливим для лінгвістики. Синтез використовує всі лінгвістичні знання про текст, що підлягає обробці, тоді як аналіз використовує як суто лінгвістичні, так і екстралінгвістичні знання, чи то енциклопедична інформація про світ, чи інформація про буденну ситуацію. Ось чому аналіз іноді можливий на основі часткового мовного знання. Це може бути проілюстровано тим, що ми іноді можемо прочитати статтю майже невідомою мовою, якщо галузь дослідження й тематика статті нам добре відомі. У такий спосіб ми активно використовуємо наші екстралінгвальні знання. Однак аналіз тексту вважається важливішим для сучасних додатків. Ось чому підхід генеративної граматики робить особливий акцент на аналізі, тоді як для синтезу пропонуються окремі теорії. Модель «Смисл ↔ Текст» допускає

окремий опис для аналізу, але постулює, що вона повинна містити повну лінгвістичну та будь-яку додаткову екстралінгвістичну частину.

Багаторівневий характер моделі. Модель прямо вводить збільшену кількість рівнів у мові: текстовий, два морфологічні (поверхневий та глибокий), два синтаксичні (поверхневий та глибокий) та семантичний. Представлення одного рівня вважається еквівалентним представництву будь-якого іншого рівня. Рівномірний смисл – Текстовий процесор та протилежний текстовий – Сенсорний процесор розбиті на кілька часткових модулів, що перетворюють дані з одного рівня на сусідній. Кожен проміжний рівень представляє результати роботи одного модуля і, водночас, введення іншого модуля. Поділ моделі на кілька модулів повинен спростити правила міжрівневих перетворень.

Посилений характер, що зберігає інформацію. Правила відповідності між вхідними та вихідними даними для модулів у теорії «Смисл ↔ Текст» повністю зберігають еквівалентність інформації на всіх мовних рівнях.

Різноманітність структур і формалізмів. Кожен модуль має свої правила та формалізми в теорії «Смисл ↔ Текст» через значну різноманітність структур, що відображають дані на різних рівнях (рядки, дерева та мережі, відповідно). На кожному рівні теорія «Смисл ↔ Текст» враховує лише мінімально можливий набір описових ознак. Навпаки, традиція генеративної граматики намагається знайти якийсь спільний формалізм, що охоплює всю мову, так що загальна множинність ознак різних рівнів розглядається спільно, без явного поділу на різні рівні.

Особливості глибинного та поверхневого синтаксису. Сутності та синтаксичні особливості цих двох рівнів суттєво відрізняються в теорії «Смисл ↔ Текст». Допоміжні та функційні слова поверхні зникають на глибині. Аналогічно, деякі синтаксичні характеристики словоформ присутні лише на поверхні (наприклад, ознаки узгодження роду та числа для іспанських прикметників), тоді як інші ознаки, маючи на увазі значення, зберігаються і на більш глибоких рівнях (наприклад, число для іменників). Таке розділення сприяє мінімізації описових засобів на кожному рівні. Поняття глибинного та

поверхневого синтаксичного рівнів у теорії Хомського також, але, як ми вже могли бачити, вони визначаються там зовсім по-іншому.

Незалежність між синтаксичною ієрархією слів та їх порядком у реченні. Ці два аспекти речення, позначені деревом залежностей, та порядок слів, передбачаються різними, хоча взаємопов'язаними чинниками. Формально це призводить до систематичного використання граматик залежностей на синтаксичному рівні, а не граматик виборчих округів. Тому основні правила міжрівневих перетворень виявились зовсім іншими в теорії «Смисл ↔ Текст», порівняно з генеративною граматикою. Основна перевага граматик залежностей вбачається в тому, що зв'язки між значущими словами зберігаються на семантичному рівні, тоді як для граматик виборчих округів (за винятком HPSG) семантичні зв'язки повинні розкриватися за допомогою окремого механізму.

Орієнтація на мови, що відрізняються від англійської. Певною мірою протиставлення граматики залежності та вибіркових груп пов'язане з різними типами мов. Граматики залежності особливо підходять для мов із вільним порядком слів, таких як латинська, російська чи іспанська, тоді як граматики вибіркових округів підходять для мов із усталеним порядком слів, таких як англійська. Однак модель «Смисл ↔ Текст» підходить для опису таких мов, як англійська, французька та німецька. Величезний досвід роботи з деревами залежностей накопичений у рамках моделі «Смисл ↔ Текст» для декількох мов. Генеративна традиція (наприклад, HPSG) також переходить до дерев залежностей, але з певними застереженнями та якимись опосередкованими способами.

Засоби лексичних функцій та синонімічні варіації. Тільки модель «Смисл ↔ Текст» зазначала, що більша частина словосполучень, відомих у будь-якій мові, виробляється відповідно до їх взаємних лексичних обмежень. Наприклад, ми можна сказати *серцевий напад* та *сердечні привітання*, але ні *сердечний напад*, ні *серцеве привітання*, бо значення поєднуваних лексем не дозволяють усіх цих поєднань. Такі обмеження у поєднанні сформували

числення так званих лексичних функцій у рамках моделі «Смисл ↔ Текст». Обчислення включає правила перетворення синтаксичних дерев, що містять лексичні функції, з однієї форми в іншу. Людина може передати одне і те ж значення різними способами. Лексичні функції дозволяють робити ці перетворення цілком формально, реалізуючи таким чином механізм синонімічних варіацій. Ця властивість відіграє важливу роль у синтезі і не має аналога в генеративній традиції. У перекладі з однієї мови іншою, варіант, реалізований для конкретної конструкції, шукається цільовою мовою серед синонімічних синтаксичних варіантів. Лексичні функції дають змогу також стандартизувати семантичне представлення, зменшуючи різноманітність міток для семантичних вузлів.

Моделі керування. На відміну від підкатегорізаційних систем генеративної лінгвістики, керування в моделі «Смисл ↔ Текст» безпосередньо пов'язує семантичну та синтаксичну валентність слів. Не тільки дієслова, але й інші частини мови описуються з погляду моделей керування. Отже, вони дозволяють чітко вказати, як кожна семантична валентність може бути представлена на синтаксичному рівні: лише іменником, даним прийменником та іменником, будь-яким із поданих прийменників та іменником, інфінітивом або будь-яким іншим шлях. Порядок слів не зафіксований у моделях керування. Навпаки, рамки підкатегорії дієслів, як правило, зводяться лише до переліку всіх можливих комбінацій синтаксичних валентностей, окремо для кожного можливого порядку у реченні. У мовах із досить вільним порядком слів кількість таких кадрів для конкретних дієслів може сягати кількох десятків, і це затьмарює всю картину семантичних валентностей. Крім того, різноманітність наборів дієслів з однаковою комбінацією кадрів підкатегорії може бути цілком порівнянною із загальною кількістю дієслів таких мов, як іспанська, французька чи українська.

Дотримання традицій і термінології класичного мовознавства. Модель «Смисл ↔ Текст» ставиться до спадщини класичної лінгвістики набагато ретельніше, ніж генеративна обчислювальна лінгвістика. У своєму

тривалому розвитку модель «Смисл ↔ Текст» показала, що навіть підвищена точність опису та необхідність суворих формалізмів зазвичай дають змогу зберегти наявну термінологію, можливо, після надання суворіших визначень термінам. Збереглися поняття фонеми, морфеми, морфи, грамеми, лексеми, частини мови, згоди, числа, роду, часу, особи, синтаксичного суб'єкта, синтаксичного об'єкта, синтаксичного присудка, актанта, сирконстанти тощо. У рамках генеративної лінгвістики теорії іноді будуються майже з нуля, без спроб інтерпретувати відповідні явища термінами, вже відомими в загальній лінгвістиці. Ці теорії іноді ігнорували поняття й методи класичної лінгвістики, в тому числі і структуралізму. Це не завжди дає додаткову строгість. Частіше це призводить до термінологічної плутанини, оскільки фахівці в суміжних сферах просто не розуміють один одного [16, с. 137-141].

1.2.5. Завдання лінгвістичних моделей

У сучасній теоретичній лінгвістиці деякі дослідники вивчають фонологію, інші морфологію, треті синтаксис, а четверті семантику та прагматику. У фонології хтось поглинувся акцентуацією, семантикою, мовленнєвими актами і т. ін.

Основними критеріями істинності в теоретичних лінгвістичних дослідженнях є його логічний характер, узгодженість та відповідність між інтуїтивними уявленнями про дані мовні явища автора теорії та інших членів спільноти лінгвістів.

У цьому сенсі праці сучасних фахівців з теоретичного мовознавства здаються лише етапами внутрішнього розвитку цієї науки. Часто здається непотрібним класифікувати їх відповідно до того, підтримують вони чи відповідають якійсь повній моделі.

Ситуація в обчислювальній лінгвістиці дещо інша. Тут критерієм істинності є близькість результатів функціонування програми для обробки мовних висловлювань до ідеального результату, що визначається розумовими здібностями середнього носія мови. Оскільки процедуру обробки через її складність слід розділити на декілька етапів, цілком необхідна повна модель,

щоб рекомендувати, які формальні ознаки та структури слід призначати висловлюванням та мові в цілому на кожному етапі, і як ці особливості повинні взаємодіяти та брати участь на кожному етапі лінгвістичних перетворень в комп'ютері. Таким чином, усі теоретичні передумови та результати мають бути подані тут досить чітко і повинні відповідати один одному у своїх структурах та інтерфейсах.

Теоретики розповідають нам про піднесення експериментальної лінгвістики на цій основі. Здається, що в майбутньому експериментальні випробування найглибших результатів усіх «часткових» лінгвістичних теорій стануть неминучим елементом еволюції цієї науки в цілому. Щодо обчислювальної лінгвістики, то комп'ютеризоване експериментування зараз є вирішальним, і на нього безпосередньо впливає те, які структури обрані для опису мови та які етапи обробки рекомендує теорія.

Тому, на перший погляд, філософська проблема лінгвістичного моделювання виявилася первісною для обчислювальної лінгвістики [16, с. 143-145].

1.3. Морфологічний та синтаксичний виміри

Завдяки комп'ютерному аналізу, який здійснює різні складні математичні обчислення, сьогодні існує вулика кількість систем обробки мовлення на всіх її рівнях. Якість таких процесів залежить від дослідницьких моделей, створених для тієї чи тієї мови, оскільки програма потребує навчання саме на формальних структурованих даних.

Багато невирішених завдань, або частково вирішених, для української мови постає на рівні синтаксичного та семантичного аналізів. У нашій роботі детальніше розглянемо синтаксичну обробку мовлення, а саме граматичну структуру речень, оскільки вона поєднує два взаємопов'язаних рівні: морфологічний і синтаксичний.

1.3.1. Морфологічний рівень

Синтаксичний розбір є стандартною технікою, яка використовується в галузі обробки природної мови. Але перш ніж синтаксичний синтаксичний

аналізатор зможе розібрати речення, йому необхідно надати інформацію про кожне слово в реченні. Наприклад, щоб розібрати речення *Дівчина – координатор проекту*, синтаксичний аналізатор повинен знати, що *дівчина* – це іменник однини називного відмінка, *проекту* – іменник однини у родовому відмінку тощо. Таку інформацію може надати словник, який просто перераховує всі словоформи з їх частиномовною належністю та флективною інформацією, наприклад, числом і часом. Українська мова має складну флективну систему, оскільки є синтетичною мовою. Керуватись ми можемо словником Є.А. Карпіловської Кореневий гніздовий словник української мови» або ж Електронним граматичним словником української літературної мови. Тому для таких мов потрібно створити аналізатор слів, який використовуватиме морфологічну систему мови для визначення частини мови та флективних категорій будь-якого слова.

Слова можуть складатися з кореня (що несе основне словникове значення) й одного або кількох афіксів, що містять граматичну інформацію. Наприклад: Cats walks smoothly – cat+N+PL walk+V+PresPart smooth+Adj+Sup Морфологічний розбір – це проблема виділення лексичної форми з поверхневої форми (Для обробки мовлення воно включає визначення меж слів.) Ми повинні враховувати: аналітичні форми (наприклад, писатиму → буду писати) Систематичні правила (наприклад, у іменника родового відмінка однини жіночого роду першої відміни забирається закінчення -а(-я) й додається -і: дівчина → давчині, людина → людині).

Українська мова, як і інші слов'янські мови, що, має складну та продуктивну деривaційну морфологію. Наприклад, від кореня *зерн* походять такі похідні форми, як *зерно, зернина, зернятко, зерновий, зернистий, зернитися* тощо. Дуже важка перерахувати в словнику всі похідні форми (включно з новими термінами чи стилістично забарвленими вигаданими словами), які можуть траплятися в природному тексті.

Дворівнева модель морфології. Значний прорив у галузі морфологічного розбору відбувся в 1983 році, коли Кіммо Коскенніні, фінський

вчений, написав дисертацію «Дворівнева морфологія: загальна обчислювальна модель для розпізнавання та генерації словоформ». Модель дворівневої морфології Коскенніні базувалася на традиційному розрізненні, яке лінгвісти проводять між морфотактикою, що перераховує перелік морфем і вказує, в якому порядку вони можуть траплятися, і морфофонемікою, яка пояснює альтернативні форми або «написи» морфем відповідно до фонологічний контекст, у якому вони трапляються. Наприклад, слово *руці* аналізується морфотактично як основа *рук*, від якої забирається закінчення *-а* й додається *-і*. Проте відбувається чергування приголосних -к-/-ц- аким чином *рук-* і *руц-* є аломорфами або альтернативними формами однієї і тієї ж морфеми. Модель Коскенніні є «дворівневою». Наприклад, слово *руці* має таке дворівневе представлення (де + – це морфемний символ межі, а 0 – нульовий символ): рука + -і → ру 0 -ці.

Незабаром після появи дисертації Коскенніні Лаурі Карттунен та інші створили LISP-реалізацію дворівневої моделі Коскенніні і назвали її KIMMO. Вона мала два аналітичні компоненти: компонент правил і лексичний компонент, або лексикон. По-перше, компонент правил складався з дворівневих правил, які враховували регулярні фонологічні або орфографічні чергування. По-друге, у лексиконі перераховані всі морфеми (основи та афікси) у їхній лексичній формі та визначені морфотаксичні обмеження. Наприклад, лексикон включав би лексичні записи для основи дієслова chase і суфікса -ed і вказував би їх відносний порядок. Використовували ці компоненти даних дві функції обробки, генератор і розпізнавач. Генератор приймає як вхідну лексичну форму, таку як spy+s, і повертає поверхневу форму spies. Розпізнавач прийме як вхідну форму поверхневу форму, таку як spies, і поверне базову форму, розділену на морфеми, а саме spy+s і додаткову інформацію , наприклад N+PLURAL.

У 1990 році Літній інститут лінгвістики випустив першу версію PC-KIMMO, реалізацію дворівневої моделі, яка точно наслідувала KIMMO Картунена. Написаний на мові програмування C, він працював на таких

персональних комп'ютерах, як IBM PC, Macintosh, а також UNIX. PC-KIMMO був досить добрим у тому, для чого він був розроблений – токенізувати слово в послідовність позначених морфем. Але у нього був серйозний недолік: він не міг безпосередньо визначити частину мови слова чи його флективні категорії.

У 1993 році була розроблена друга версія PC-KIMMO. До неї додали модуль з граматикою слів. Слово граматика – це синтаксичний аналізатор діаграм на основі уніфікації, який забезпечує розбір похідних і структурних елементів. Спочатку синтаксичний аналізатор діаграм був розроблений для синтаксичного аналізу. Подібно до того, як синтаксичний аналізатор речень створює дерево синтаксичного аналізу зі словами як його листкові вузли, синтаксичний аналізатор слів створює дерево розбору з морфемами як його листкові вузли. Коли ви робите синтаксичний розбір речення, воно зазвичай вже перетворюється на слова. Ця токенізація здійснюється за правилами та за словником. Коли перше слово подається до PC-KIMMO Recognizer, правила та словник аналізують слово на послідовність морфемних структур (або, можливо, більше однієї послідовності, якщо знайдено більше одного аналізу). Морфемна структура складається з лексичної форми, її моделі, категорії та ознак (рис. 1.1, 1.2) [1]:

```
Form:   en+          large          +ment          +s
Gloss:  VR1+         AJ             +NR25          +PL
Cat:    PREFIX       ROOT           SUFFIX         INFL
Feat:   [fromcat: AJ  [lexcat: AJ    [fromcat: V    [fromcat: N
         tocat: V      aform: !POS]   tocat: N       tocat: N
         finite: !-]                  number: !SG]   number: SG
                                                      reg: +]
```

Рисунок 1.1 – Послідовність морфемних структур

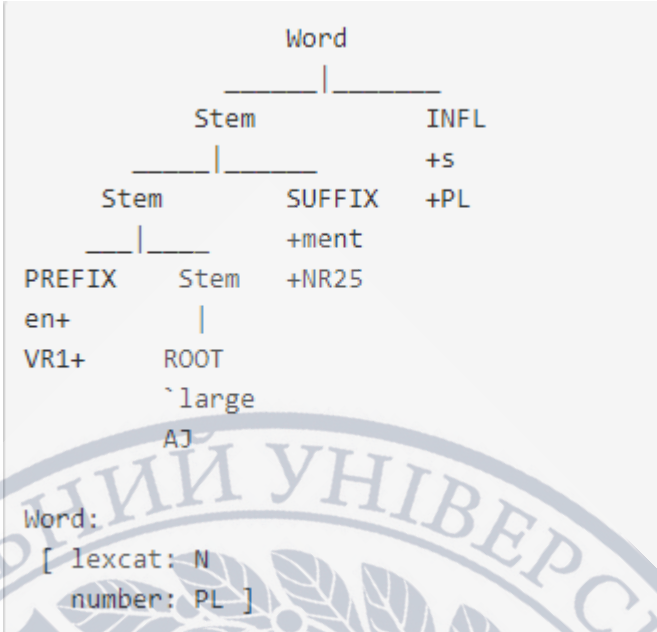

Рисунок 1.2 – Дерево розбору

В Україні алгоритм і програму лематизації російської мови розробила В. Перебийніс. У роботі зведення парадигм російської мови базується на аналізі кінцівок словоформ. Укладено два алгоритми: для тексту, у якому кожному слововживанню присвоюється код граматичного класу (частини мови) і граматичного підкласу (рід, число, відмінок, особа, час), і для нерозміченого тексту.

Зведення парадигм включає в себе кілька часткових задач, які розв'язуються у такій послідовності:

- виявлення флексії і відокремлення її від основи слова;
- виявлення варіантів основи, якщо такі є;
- об'єднання всіх форм слова в одну групу (парадигму);
- виділення або реконструкція словникової форми слова.

Оскільки кожний граматичний клас слів, який має словозміну, має і свої особливості формоутворення й об'єднання форм у парадигми, то в алгоритмі доцільно передбачити таку кількість блоків, яка відповідає кількості граматичних класів, що характеризуються словозміною. Відповідно до прийнятого переліком граматичних класів слів виділяються такі блоки:

• зведення парадигм іменників (чоловічого, жіночого, середнього родів, pluralia tantum);

• зведення парадигм атрибутивних класів (прикметників, дієприкметників, порядкових числівників);

• зведення дієслівних парадигм;

• зведення парадигм предикативних класів (скорочених форм прикметників і дієприкметників);

• зведення займенникових парадигм (займенників- іменників, займенників-прикметників);

• зведення парадигм кількісного числівника;

• зведення варіативних форм прийменників.

Алгоритм працює як на однозначно встановлених, так і на омонімічних (диз'юнктивних) класах, тобто до початку роботи контекстного аналізу. У міру зняття омонімії граматичних класів слів код диз'юнктивного класу заміняється однозначним кодом, але на результати виділення основи та флексії це не впливає, хоча парадигма диз'юнктивного класу може бути розділена на дві, тобто групування словоформ у парадигму зміниться [3, с. 55-56].

Отже, для того, щоб побудувати морфологічний аналізатор, нам потрібно буде наступне:

1. Лексика: список основ та афіксів, разом із основною інформацією про них (чи ця основа є іменною, чи дієслівною тощо).

2. Морфотактика: модель порядку морфем у слові, яка пояснює, які класи морфем можуть слідувати іншими класами всередині слова. Наприклад, існує правило, що в морфема множини слідує за іменником, а не перед ним.

3. Орфографічні правила: ці правила правопису використовуються для позначення зміни, що відбуваються у слові, зазвичай при поєднанні двох морфем (наприклад, y → ie, правило написання яких зазначено вище, змінює city + -s на cities, а не citys) [26, с. 57-91].

1.3.2. Синтаксичний рівень

Для того, щоб комп'ютерна система розпізнавала запити, які надає їй користувач, потрібно здійснити перетворення природної, зрозумілої для людини, мови у формалізований вигляд. Цей процес відбувається завдяки роботі лінгвістичного процесора, який створює модель мовної системи [41, с. 250]. Процес його виконання є комплексним і містить такі рівні аналізу:

1. Морфологічний;
2. Синтаксичний;
3. Семантичний.

Кожний етап є фундаментом для роботи наступного. Таким чином виділяються проміжні рівні аналізу тексту – **лексико-граматичний** та **семантико-синтаксичний**.

Хоча для цих етапів розробляються окремі алгоритми, проте усі вони тісно взаємопов'язані. Ядерним постає синтаксичний аналіз. Кодовані елементи нижчого рівня (морфологічного) є базою, основними засобами для роботи алгоритмів синтаксичного аналізу. Будування зв'язків між тими чи тими елементами здійснюються з опертям на семантичний аспект аналізу [4, с. 3-5]. Результатом машинного синтаксичного аналізу має стати автоматичне визначення структури елементів речення та зв'язків між ними [3, с. 134-135].

Багато нових ідей, які використовують для розробки автоматичного синтаксичного аналізу, висловили представники дескриптивної школи структурної лінгвістики. Не будучи по суті теорією мови у звичному для неї смислі цього слова, вона дає уявлення про лінгвістичний опис як набір процедур опрацювання тексту, виконання яких у певному порядку має привести до створення моделі мовленнєвої діяльності. Саме дескриптивна гілка структурної лінгвістики поклала початок створенню моделей, що імітують дослідницьку діяльність лінгвіста: із суми спостережень над текстом лінгвіст здобуває первинне уявлення про спосіб організації тексту й у вигляді чітких процедур – правил алгоритму – сповіщає автомату свої дії, а потім з його допомогою одержує на більшому матеріалі дані, які його цікавлять. З метою

досягнення максимально об’єктивного аналізу мови, намагаючись розробити всебічну і чітку техніку лінгвістичного дослідження, представники дескриптивної лінгвістики скерували свою увагу на зовнішню мовну форму, лінгвістичні одиниці ототожнювалися і класифікувалися не на основі значення, як це мало місце у традиційній граматиці, а на основі розподілу (дистрибуції) у мовленні. При дистрибутивному аналізі кожне явище вивчається в оточенні інших явищ й у взаємодії з іншими явищами [4, с. 97].

У 1950-х роках, коли почалася комп’ютерна ера, видатний лінгвіст Ноам Хомський розробив деякі нові формальні засоби, спрямовані на кращий опис фактів різними мовами.

Серед формальних інструментів, розроблених Хомським та його послідовниками, можна виділити два найважливіші компоненти:

- чисто математичне ядро, яке включає генеративну граматику, розташоване в ієрархії граматики різної складності. Генеративна граматика виробляє рядки символів, і набори цих рядків називаються формальними мовами, тоді як в загального мовознавства їх можна було б назвати текстами. Граматики безпосередніх зв’язків становлять один рівень цієї ієрархії;
- спроби описати низку штучних і природних мов в рамках граматик безпосередніх зв’язків. Фразові структури були формалізовані як контекстно-вільні граматики і стали основним інструментом для опису природних мов, в першу чергу, англійською [16, с. 35].

У сучасній лінгвістиці існує два підходи до реалізації автоматичного синтаксичного аналізу через моделювання структур речення – з використанням 1) **граматики безпосередніх зв’язків (граматики безпосередніх складових)** та 2) **граматики залежностей**. Перший спосіб полягає в передаванні реченнєвої структури через бінарні зв’язки (головної частини та підпорядкованої):

{(Сонце) [світить (високо вгорі)]}.

Наведене речення складається з таких елементів: іменних частин – *сонце, високо вгорі* та дієслівної частини – *світить високо вгорі*. Цю модель можна зобразити також за допомогою синтаксичного дерева залежностей (рис. 1.1):

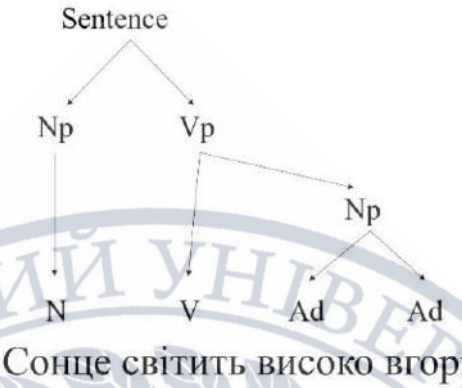

Рис. 1.1 – Дерево безпосередньо складових

Метод БС заснований на таких припущеннях:

- істотну роль відіграє лише одне відношення – відношення підрядності;
- речення будується шляхом послідовного поєднання одне з одним якихось дрібних його складників: спочатку слова поєднуються одне з одним, утворюючи певний складник, потім цей складник поєднується з іншим складником або окремим словом, й утворюється новий складник тощо. Останній акт побудови речення – це поєднання груп підмета і присудка. Аналіз продовжується доти, доки все речення не буде представлене у вигляді єдиного блоку, оскільки відношення підрядності бінарне, кожний блок складається не більше, ніж із двох частин [4, 97-98].

Для створення схеми простих речень метод використовує початковий символ S створюваного речення й декілька інших нетермінальних символів: символ фрази іменника NP, символ дієслівної фрази VP, символ іменника N, символ дієслова V, символ детермінанта D. Усі ці нетермінальні символи інтерпретуються як граматичні категорії. Набір правил може бути таким:

$S \rightarrow NP\ VP$

$VP \rightarrow V\ NP$

$NP \rightarrow D\ N$

$NP \rightarrow N$

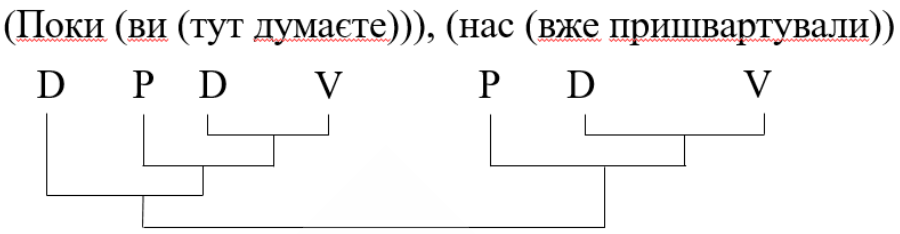

Рис. 1.2 – Ієрархія зв’язків

Кожен символ у правій частині правила вважається складником символізованої ліворуч сутності. Використовуючи ці правила в будь-якому можливому порядку, ми можемо перетворити S у рядки D N V D N, або D N VN, або N V D N, або N V N тощо.

Додатковий набір правил використовують для перетворення всіх цих нетермінальних символів у термінальні, що відповідають заданим граматичним категоріям. Термінали – це звичайні слова будь-якої мови, які допускають ті самі категорії та той самий порядок слів:

N – інгредієнти, тональності, побутом.

V – приніс, з’явився, пофарбую.

D – сюди, там, досі.

На початковому етапі розробки генеративного підходу виникла ідея незалежного синтаксису, а проблема обробки природної мови розглядалася як визначення синтаксичної структури кожного речення, що формує текст. Синтаксичну структуру речення ототожнюють зі структурою, що поділяє речення на частини, потім ці частини на менші частини та ін. Ця композиція відповідає послідовності застосувань граматичних правил, які генерують дане речення.

Подальші дослідження стали використовуватися не тільки для опису природних мов, а й для специфікації формальних мов, наприклад тих, що використовуються в математичній логіці, розпізнаванні образів і в мовах програмування. З цих досліджень виникла нова галузь науки під назвою математична лінгвістика.

Протягом наступних трьох десятиліть після піднесення математичної лінгвістики багато зусиль було присвячено вдосконаленню її інструментів, щоб вона краще відповідала фактам природних мов. На початку ці дослідження випливали з основних ідей Хомського та були дуже близькі до них.

Однак незабаром стало очевидним, що безпосереднє застосування простих безконтекстних граматик до опису природних мов стикається з великими труднощами. Під тиском суто лінгвістичних фактів і з метою краще пристосувати формальні інструменти до природних мов Хомський запропонував так звані трансформаційні граматики. В основному вони були орієнтовані на англійську мову та пояснювали, як побудувати питальне чи заперечне речення з відповідного стверджувального та як трансформувати речення активного мовлення в його пасивний еквівалент тощо [16, с. 35-39].

Наступним способом представлення формальної структури речень є граматики залежностей. Якщо граматики безпосередніх зв'язків зображують структуру речення через групи слів, то граматики залежностей будують синтаксичну модель з аналізом кожного слова окремо (рис. 2). Граматики залежностей підпорядковуються правилам контекстно-вільної граматики, де кожне залежне слово будує зв'язок лише з одним головним (означення підпорядковуються означуваному, додатки – дієсловам, іменники прийменникам та ін.). У такій моделі головним незалежним елементом вважається дієслово-присудок, саме воно є вершиною речення [3, с. 135-138; 26, с. 322-328], пор.: *Делегація молодих науковців приїжджає завтра*:

Рис. 1.3 – Схема речення граматики залежностей

У цих граматиках залежності складники та правила фразової структури не відіграють жодної фундаментальної ролі. Натомість синтаксична структура речення описується винятково в термінах слів і бінарних семантичних або синтаксичних відносин між цими словами (так звані лексичні залежності).

Дослідники граматики залежностей стверджують, що однією з головних переваг чистих граматик залежностей є їх здатність обробляти мови з відносно вільним порядком слів. Наприклад, порядок слів у таких мовах, як українська, набагато гнучкіший, ніж в англійській; об'єкт може поставати до або після прислівника. Граматика фразової структури потребує окремого правила для кожного можливого місця в дереві розбору, щоб таке прислівникове словосполучення могло виникнути. Граматика залежностей матиме лише один тип покликання, що представляє це конкретне прислівникове відношення. Таким чином, граматика залежності абстрагується від варіацій порядку слів, представляючи лише ту інформацію, яка необхідна для розбору [26, с. 459–462].

Граматика залежностей розбудована на дослідженнях А. М. Лешковського, Л. Теньєра, Лесерфа. Її основу становлять поняття залежності. Відповідно до нього між мінімальними синтаксичними одиницями в реченні встановлюється відношення часткової впорядкованості. Одна з одиниць вважається незалежною, решта всіх залежить від певної одиниці та обов'язково тільки від однієї (одиниця, від якої залежать інші — господар, залежна одиниця — слуга). Структура пропозиції, яка визначається сукупністю зв’язків в реченні між його компонентами, формально може бути зображена в похідних дерева залежностей. Дерево залежностей є упорядкованим спрямованим графом, організованим таким чином, що головній вершині його, якій безпосередньо чи опосередковано підпорядковуються всі інші, відповідає незалежний елемент пропозиції. Ребра графа задаються стрілками, спрямованими від господаря до слуги. У кожну вершину графа може входити лише одна стрілка. Кількість стрілок, що виходять з вершини, необмежена. існує ієрархічний зв'язок між вузлами дерева. Створення реальної граматики залежностей, призначеної для

використання в синтаксичному аналізі реальних текстів, вимагає лінгвістичної інтерпретації, насамперед, поняття «мінімальної синтаксичної одиниці» і поняття «зв'язку залежності».

У Л. Теньєра, А. М. Лешковського та в низці систем машинного перекладу як мінімальну синтаксичну одиницю обирають члени речення, оскільки саме вони, маючи структурно-синтаксичне значення, є носіями синтаксичної функції. Таке виділення мінімальної синтаксичної одиниці відповідає розумінню синтаксичного зв'язку, який завжди є функційним. Але у практичній реалізації граматик залежностей в автоматичному синтаксичному аналізі, що керує членами речень, виникають труднощі, пов’язані зі складністю алгоритмічної процедури встановлення меж членів речення, що складаються з кількох слів, по-друге, з неможливістю формально здійснити функційну диференціацію залежності членів речення, що підпорядковуються одному складному члену [4, с 6-8].

Ще однією важливою теорією, яка з’явилася у зв’язку з поняттям залежності, була теорія валентностей слова. Цей термін було введено у лінгвістику в 1948 р. російським дослідником С. Кацнельсоном, який визначив це поняття як «властивість слова певним чином реалізуватися в реченні і вступати в певні комбінації з іншими словами». У дослідженнях А. Теньєра також розроблене поняття валентності, яку він визначає як здатність дієслова керувати певним числом актантів (від 0 до 3). І якщо Теньєр обмежував розуміння валентності кількісною стороною, то послідовники вже звертали увагу і на її якісний аспект: морфологічні, конструктивні, семантичні особливості головного і залежного компонента. У російській лінгвістичній школі поштовхом для подальшого розвитку цієї теорії послугувала розробка алгоритмів автоматичного перекладу, які будуються за принципом пошуку елементів, які «насичують валентність» головного слова (Г. Цейтін, А. Засоріна).

Поняття валентності почали розуміти як потенційний зв’язок одного мовного елемента (фонема, морфема, слово, словосполучення) з іншими

мовними елементами, а сполучуваність – як реальний зв’язок мовної одиниці з іншими одиницями в мовленні [5, с 107].

Для подолання труднощів зі встановленням меж членів речення пропонується обрати за мінімальну синтаксичну одиницю словоформу, представлену граматичним класом, таксономічну одиницю мови. І тоді поняття зв'язку визначатиметься через властивість таксономічної одиниці – через валентність.

Від вибору мінімальної синтаксичної одиниці залежить статус максимальної синтаксичної одиниці. У цьому випадку максимальною синтаксичною одиницею можна вважати речення без розмежування на прості та складні. Визнання словоформи мінімальною синтаксичною одиницею змушує відмовитися від ототожнення предикативних частин з членами речення, що, своєю чергою, обумовлює необхідність вибору предикативних частин поряд із простим реченням як максимальна синтаксична одиниця.

Другому альтернативному рішенню віддають перевагу майже всі відомі системи машинного перекладу. В цьому випадку при побудові синтаксичного подання складного речення прийнято зображати предикативну частину у вигляді вузлів зі спеціальною міткою, а саму структуру предикативної частини як окреме дерево, головна вершина якого підпорядковується відповідній нетермінальній вершині. Такий спосіб подання структури речення дає можливість формально відобразити кількість предикативних частин і визначити відношення залежності між ними, але не може показати спосіб включення окремих предикативних частин до складної пропозиції. У цій системі автоматичного синтаксичного аналізу для розкриття характеру зв'язку між предикативних частин і для вказівки синтаксичних засобів, що використовуються при їх організації, пропонується дерево залежностей окремих предикативних частин вводити в один узагальнений граф без підпорядкування їх штучним вершинам, зображуючи зв'язки між ними [4, с. 7-9].

На сьогодні для української мови реалізований синтаксичний модуль граматики АГАТ, розроблений лабораторією комп'ютерної лінгвістики Інституту філології Київського національного університету імені Тараса Шевченка. Його робота відбувається у два етапи:

1. Встановлення зв'язків між словами в межах словосполучень (прийом граматики безпосередніх складників).
2. Аналіз речення через побудову дерева залежностей.

Цей модуль був застосований до Електронного корпусу української мови, де речення вже можуть аналізуватись і на синтаксичному мовному рівні.

РОЗДІЛ 2.
ОСОБЛИВОСТІ ПРИРОДНОЇ ОБРОБКИ МОВИ

2.1. Основні проблеми автоматичного синтаксичного аналізу

Процес аналізу речення не завжди дає правильні результати. Існує низка проблем, які потребують детальнішого вивчення й напрацювання шляхів усунення суперечливих питань:

- омонімія й полісемія. Лексична омонімія призводить до неоднозначності на рівні типу синтаксичного зв'язку: *Він рекомендований нам інспектором*. ДЕ?;
- граматична конверсія: *«Розкопуйте **похованих** у землі сліпих велетнів»* – використання слова *похованих* у ролі іменника: *[Розкопуючи похованих] в [землі сліпих велетнів]*;
- неоднозначність інтерпретації в синтаксичних конструкціях: *Доповідь студентки, про яку я вам говорив*; ДЕ?
- валентна варіативність – факультативність сирконстант. Сирконстанти можуть знаходитись у реченні потенційного головного елемента речення, викликаючи синтаксичну неоднозначність: ***Доповідь** про пограбування в інституті соціології була цікавою*;
- валентна варіативність – варіативність актантів, тобто здатність активно валентного слова заповнювати валентності актантами різних типів, а також підпорядковувати собі різні форми: *Існує можливість прохання начальника уникнути*; ДЕ?
- валентна варіативність – факультативність актантів, яка дає змогу відносити залежний елемент як до одного, так і до іншого потенційного головного слова: *Учитель співу не чує*; І ДЕ?
- виникнення синтаксичної неоднозначності провокована такими поєднаннями:

– наявність залежного елемента, який може бути віднесений до одного або декількох однорідних членів речення: *вік атомної* ***[енергетики, автоматизації]****, століття* ***[атомної енергетики]****,* ***[автоматизації]***;

– наявність елемента, що може бути однорідний з іншим залежним елементом або словом, якому останнє підпорядковане: *Чи не настав до стану* ***[втрати [ініціативи, байдужості]]****…; Чи не настав до стану* ***[[втрати ініціативи], [байдужості]]*** *…*;

– наявність елемента, який може виступати головним стосовно декількох однорідних членів речення або тільки до одного з них, де слово, що не підпорядковане головному не є з ним однорідним: *Довіряти* ***[доньці і батьку]*** *забороняли* (конструкція з однорідними членами) – ***[Довіряти доньці] [і батьку забороняли]*** (конструкція з підсилювальним *і*);

– перерозподіл однорідності у фразах на зразок: *Мій брат або Петро й Михайло залишаться.*

Виділяються також причини, що самі собою не викликають синтаксичної неоднозначності, але сприяють її виникненню. До них можна віднести:

• еліпс. Якщо один з елементів речення відсутній, на його місце в може претендувати інший елемент, створюючи неоднозначність: *Купив яблуню і посадив у дворі сусіда*;

• імплементовані елементи (вставні слова й конструкції, дієприкметникові та дієприслівникові звороти), що розділяють залежне і головне слово: *Він йшов,* ***незважаючи на мороз****, в легкій куртці,* ***мабуть****,* ***відчуваючи себе чудово*** – є синтаксично неоднозначним саме через наявність одночасно дієприслівникових зворотів і вставного слова, які замінюють собою можливі знаки пунктуації, через що структура пропозиції стає не зовсім зрозумілою;

• порядок слів. Оскільки українській мові притаманний вільний порядок слів, то виникає велика кількість неоднозначних синтаксичних конструкцій: *Подейкували про можливий прихід військ Наливайка* [13].

2.2. Методи математичної лінгвістики

Математична лінгвістика виникла в 50-ті роки XX ст. Нині існує два різні погляди на математичну лінгвістику. На більш вузький погляд, термін математична лінгвістика використовується для теорії формальних граматик конкретного типу, що називаються генеративними граматиками. Це одна з перших чисто математичних теорій, присвячена природній мові. З іншого боку, у широкому розумінні математична лінгвістика – це перетин між лінгвістикою та математикою, тобто тією частиною математики, яка приймає лінгвістичні явища та взаємозв'язки між ними як об'єкти можливих застосувань та інтерпретацій.

Підґрунтям появи математичної лінгвістики стала необхідність в уточненні основних лінгвістичних понять, потреба в уведенні точніших та об'єктивніших методів для аналізу та синтезу мови та тексту, поява міжпредметних зв'язків з іншими галузями, що вимагають спілкування мовою математики. З розвитком можливостей комп'ютерних технологій також виникла потреба, зокрема, у машинному перекладі та автоматизованому інформаційному пошуку.

У другій половині XX століття виникла нова галузь прикладної лінгвістики, а саме обчислювальна або інженерна лінгвістика.

Обчислювальну лінгвістику можна розглядати як синонім автоматичної обробки природної мови, оскільки основним завданням обчислювальної лінгвістики є лише побудова комп'ютерних програм для обробки слів та текстів природною мовою.

Комп'ютерна лінгвістика (машинна, обчислювальна, інженерна лінгвістика) займається застосуванням комп'ютера (технологій та програм опрацювання даних) для моделювання функціонування мови в певних умовах та виконання лінгвістичних завдань, а також розробляє лінгвістичні аспекти комп'ютеризації. У широкому розумінні до комп'ютерної лінгвістики зараховують усе, що пов'язане з використанням комп'ютерів у мовознавстві.

Математична лінгвістика – галузь науки на межі мовознавства та математики, яка вивчає можливості застосування математичних методів для опису та дослідження природних і деяких штучних мов, для пояснення лінгвістичних подій.

Саме математичну лінгвістику вважають теоретичним підґрунтям прикладної лінгвістики. Підкреслюючи спільність понятєвого апарату, математичну лінгвістику іноді розділяють на галузі мовознавства та математики, а також зазначають, що в частині використання розроблених математичних моделей для опису будови природних мов математична лінгвістика належить до такої галузі досліджень, як штучний інтелект.

Математична лінгвістика бере участь у вирішенні завдань, які є актуальними не лише для мовознавства, тому уміння їх розв'язувати є вимогою часу для лінгвістів.

В даний час квантитативна лінгвістика в основному означає статистичну лінгвістику. Він надає методи прийняття рішень при обробці тексту на основі раніше зібраних статистичних даних. Одним із типів таких рішень є вирішення двозначності у фрагментах тексту, що підлягають аналізу. Інше застосування статистичних методів полягає в розшифровці текстів забутими мовами або невідомими системами письма. Як приклад, розшифровка гліфів майя була виконана в 1950-х роках Юрієм Кнорозовим, беручи до уваги статистику різних гліфів.

У лінгвістичних дослідженнях і особливо під час реалізації алгоритмів машинного послівного перекладу та інформаційного пошуку постійно виникають завдання, пов'язані з прогнозуванням появи в сегменті заданої довжини певної кількості словоформ чи словосполучень, що належать певним класам. Ймовірнісне моделювання тексту та складів, словосполучень, граматичних класів тощо дає змогу вирішити це завдання та визначати об'єм вибірки, необхідної для забезпечення із заданою ймовірністю появи хоча б один раз відповідної лінгвістичної одиниці.

Вивчення функціонування мови та мовлення за допомогою ймовірнісного моделювання тексту спирається на моделі теорії ймовірності та комбінаторику.

Прикладні задачі розв'язує математична лінгвістика:

− розроблення формальних моделей природних та штучних мов;

− вирішення питань практичної транскрипції та транслітерації;

− дешифрування невідомих писемностей;

− усний та письмовий переклад, розроблення систем автоматичного послівного і пооборотного машинного перекладу, семантичний переклад тексту;

− авторська та часова атрибуція твору;

− створення систем стенографії, систем письма для сліпих;

− завдання судової та кримінальної лінгвістики;

− розроблення раціональної та стабільної орфографії;

− автоматичне розпізнавання та синтез мови;

− розроблення автоматизованих систем опрацювання текстової інформації [13].

2.3. Особливості природної обробки мови

Природна обробка мови (англ. Natural language processing, NLP) – це міждисциплінарна галузь лінгвістики та комп'ютерних наук, дослідження якої застосовуються для розроблення штучного інтелекту (ШІ, англ. Artificial intelligence, AI), а саме – розуміння комп'ютером мовлення людини. Такі технології зазвичай інтегруються в програми, що забезпечені переведенням мовлення в текст, машинним перекладом, автоматичним анотуванням та реферуванням текстів. NLP широко використовується в бізнес-проектах для вирішення різних корпоративних завдань, наприклад, для оброблення великої кількості запитів клієнтів або для швидшого комунікування між замовником та виконавцем [29].

Для того, щоб комп'ютер зрозумів людське мовлення, потрібно надати дані, зрозумілі для його прочитання. Отже, зробити це можна за допомогою алгоритмів обробки природного мовлення, які базуються на статистичних

даних та моделях машинного навчання (machine learning) й глибокого навчання (deep learning).

Статистичні дані ми отримуємо зі словників (конкордансів, частотних словників) та корпусів текстів – маркованої сукупності текстів, зібраних за певними критеріями (тип, жанр, автор і т. ін.).

Моделі машинного навчання та глибокого навчання – це одні з найважливіших ланок штучного інтелекту. Їх різниця полягає в підходах до обробки інформації. **Машинне навчання** використовує алгоритми, що, аналізуючи структуровані дані, оброблюють вхідну інформацію. Наприклад, маємо текст, у якому потрібно визначити іменники. Попередньо необхідно створити словник частин мов, корпус текстів, у якому частини мови будуть проіндексовані. Після цього створюється алгоритм, який проаналізує тексти на основі структурованих даних (словника та корпусу текстів), тобто відбудеться навчання моделі. У такий спосіб модель буде готова знаходити іменники.

Глибоке навчання працює по-іншому. В основі обробки інформації знаходяться нейронні мережі, що за принципом виконання завдань схожі до роботи нервових клітин людини. Глибоке навчання на відміну від машинного навчання не потребує попередньої обробки структурованих даних. Якщо в систему завантажити текст і поставити програмі завдання – знайти прикметники, то програма буде працювати таким чином, що текст проходитиме через низку алгоритмів, кожен з яких відповідає за окреме завдання (розпізнати токени, віднайти частини мови, обчислити подібності/відмінності за певними критеріями). Така обробка інформації нагадує ієрархічну модель, оскільки отримані дані на виході з алгоритмів однієї ланки переходять до алгоритмів наступної. Із висновків останньої ланки отримаємо дані роботи нейронних мереж [25].

2.3.1. Моделі машинного навчання.

Для машинного навчання виділяють чотири основні моделі

- контрольоване, або навчання з учителем (англ. Supervised learning);

- неконтрольоване, або навчання без учителя (англ. Unsupervised learning);
- напівавтоматичне навчання, або часткове навчання (англ. Semi-supervised learning);
- навчання з підкріпленням (англ. Reinforcement learning) [23].

У першому випадку (з контрольованим навчанням) створюється відповідний до завдання алгоритм. Цей алгоритм має складатись з перевірки ознак про конкретний об'єкт. Для цього людині потрібно створити набори даних, які схарактеризують ті ознаки. У такий спосіб система, за допомогою «учителя» (людини), зокрема, за розміченими ознаками, вибудує свою логіку вирішення задачі. Контрольоване машинне навчання може бути присутнім в синтаксичному аналізі у формі бази з тегованими токенами й синтаксичними структурами, за мітками яких машина навчається та проводить аналіз [11, с. 1-2].

Неконтрольоване машинне навчання, або ж навчання без учителя, працює так: система отримує низку видів об'єктів та аналізує їх. Її завдання полягає в тому, щоб знайти спільні ознаки для того чи того виду та розбіжності між різними видами. Отже, програма сама для себе виділить властивості об'єктів, за якими проводитиме навчання та пізніше розпізнаватиме подібні. Наприклад, для лінгвістичного дослідження це може бути аналіз сполучень словоформ. Система має обробити велику кількість прикладів сполучень прикметника з іменником або інших частин мов й знайти закономірності їх узгодження [21].

Контрольоване навчання потребує значних ресурсів та зусиль, щоб зробити розмітку всіх аспектів явища, включно із синтаксичним аналізом; а неконтрольоване навчання є обмеженим у застосуванні, оскільки не завжди може виділити всі потрібні для оброблення ознаки. Для того, щоб уникнути проблемні питання, почали комбінувати ці явища. На початковому етапі напівавтоматичного навчання програма проводить самостійний аналіз без участі людини. Наступним кроком є розмітка того, що система не змогла

розпізнати й виділити. У такий спосіб робота машини набуває досконалішої форми [20].

Машинне навчання з підкріпленням полягає в послідовному опрацюванні даних. Програма аналізує вихідні дані та робить висновки (позитивні чи негативні), приймає рішення й будує алгоритм опрацювання наступної вхідної інформації. Робота цього способу є циклічною з вдосконаленням функціювання системи [15].

Існує багато алгоритмів, розроблених згідно чотирьох основних моделей машинного навчання. Для вибору відповідного алгоритму потрібно враховувати певні критерії, зокрема:

– розмір, якість та характер інформації;

– обмеження в часі;

– наскільки терміновим є завдання;

– ціль використання даних та результатів [19].

2.3.2. Алгоритми машинного навчання

Алгоритми машинного навчання – програми, які самовдосконалюються, використовуючи велику кількість даних. Навчальна частина цих програм ґрунтується на тому, що з входженням нової інформації вони можуть змінювати спосіб оброблення даних залежно від того, наскільки це потрібно для виведення правильної відповіді. Таким чином, алгоритм машинного навчання – це програма з певним способом коригування власних параметрів, враховуючи відгуки про попередні показники продуктивності, наперед прогнозуючи набір даних.

За характером моделей машинного навчання існують такі базові алгоритми:

– *контрольоване машинне навчання:*

- **Лінійна регресія** є простим алгоритмом. У машинному навчанні ми маємо набір змінних вхідних даних – x, які використовуються для визначення

вихідної змінної – y. Існує залежність між вхідними змінними та вихідною змінною. Мета машинного навчання – кількісна оцінка цих відношень.

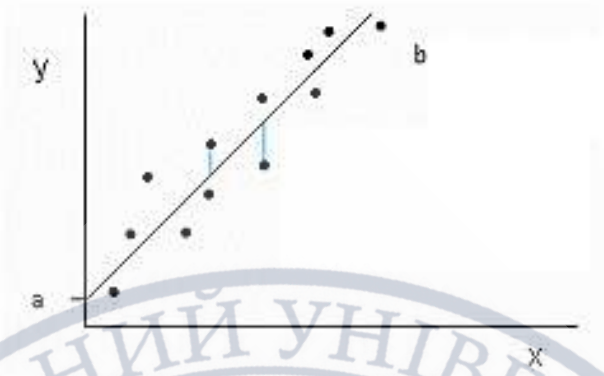

Рисунок 2.1. – Лінійна регресія

У лінійній регресії співвідношення між вхідними змінними (x) і вихідною змінною (y) виражається як рівняння: y = a + bx. Таким чином, метою лінійної регресії є з'ясування значень коефіцієнтів a і b. В нашому прикладі a – перетин, b – нахил лінії, виражає взаємозв'язок між x і y.

На рис. 1 показані графічні значення x і y для набору даних. Метою завдання є встановити лінію, найближчу до більшості точок, що зменшило б відстань між значенням точок даних від (y) та лінією.

Лінійну регресію використовують для сталих значень, наприклад, кількість опадів (см).

Цей алгоритм є простим у виконанні, але є обмеженим у використанні: не розповсюджується на функції нелінійних гіпотез. Якщо завдання складається з великої кількості даних, то використання цього алгоритму не є продуктивним з огляду на затрати часу.

• **Логістична регресія.** Прогнози логістичної регресії – це дискретні значення, які можна отримати після застосування функції перетворення. Логістична регресія найкраще підходить для двійкової класифікації, в якій набори даних такі: y = 0 або y = 1, де 1 позначає клас за замовчуванням. Наприклад, при прогнозуванні, чи відбудеться подія чи ні, є лише два варіанти: вона відбудеться (1) або ні (0). Отже, якщо зробити припущення, що людина є хворою, в нашому наборі даних використовується значення – 1.

Логістична регресія отримала назву від функції перетворення, що вона використовує, яка й називається логістичною функцією: h (x) = 1 / (1 + ex). Її результатом є S-подібна крива.

У логістичній регресії вихід має форму ймовірностей класу даних за замовчуванням (на відміну від лінійної регресії, де вихід безпосередньо продукується). Оскільки це є ймовірність, то результат ми можемо отримати в межах від 0 до 1. Наприклад, якщо завданням є визначити, чи пацієнт є хворим, а ми вже знаємо, що хворих пацієнтів позначають як 1, тоді якщо алгоритм дає пацієнту оцінку 0,98, він вважає, що пацієнт, швидше за все, хворий.

Вихідні дані (y-значення) генеруються за допомогою журналу, що перетворює значення x, використовуючи логістичну функцію h (x) = 1 / (1 + e^ - x). Потім ці дані співставляються, щоб перетворити ймовірність у бінарну класифікацію.

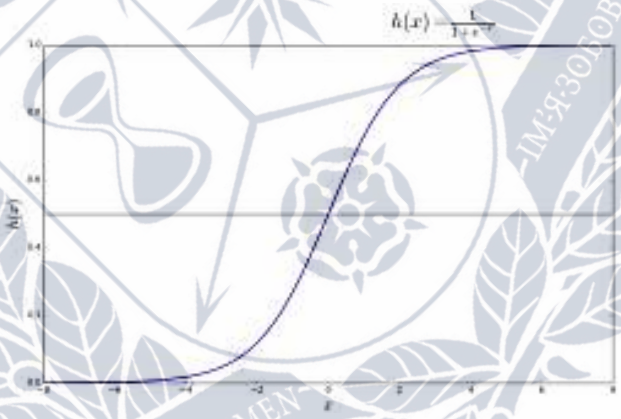

Рисунок 2.2. – Логістична регресія

- **Дерево класифікацій (Дерево регресій)** має ієрархічну форму й складається з вузлів – некінцевих та кінцевих. Некінцеві самі по собі класифікуються на кореневий й внутрішні вузли. Кожен некінцевий вузол являє собою одну вхідну змінну (x) та точку розщеплення на цій змінній; кінцеві вузли є визначають вихідну змінну (y). Дерева рішень можуть використовуватися для класифікації даних, і вони включають певні припущення того, яким може / не може бути конкретний екземпляр, аналізуючи особливості варіантів вихідної інформації. Воно більше, ніж лисиця? Так, тоді це не заєць. Це живе? Так, тоді це не велосипед.

Цю модель легко побудувати, і вона може працювати з великими наборами даних, підготовка яких зазвичай не є трудомісткою. Водночас робота алгоритму потребує великих обчислювальних ресурсів та часу при наявності великої кількості даних.

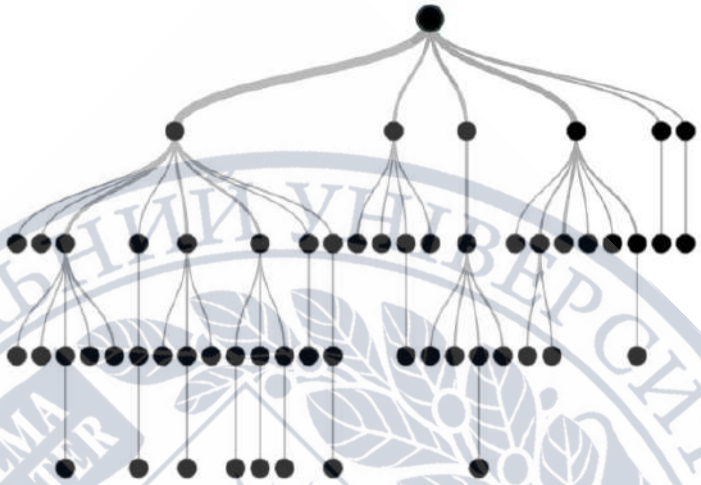

Рисунок 2.3. – Дерево рішень

• **Random Forest** складається з багатьох дерев-рішень. Вони являють собою цілісну систему. Кожне дерево рішень створене за допомогою підмножини атрибутів, які використовуються для класифікації певної сукупності.

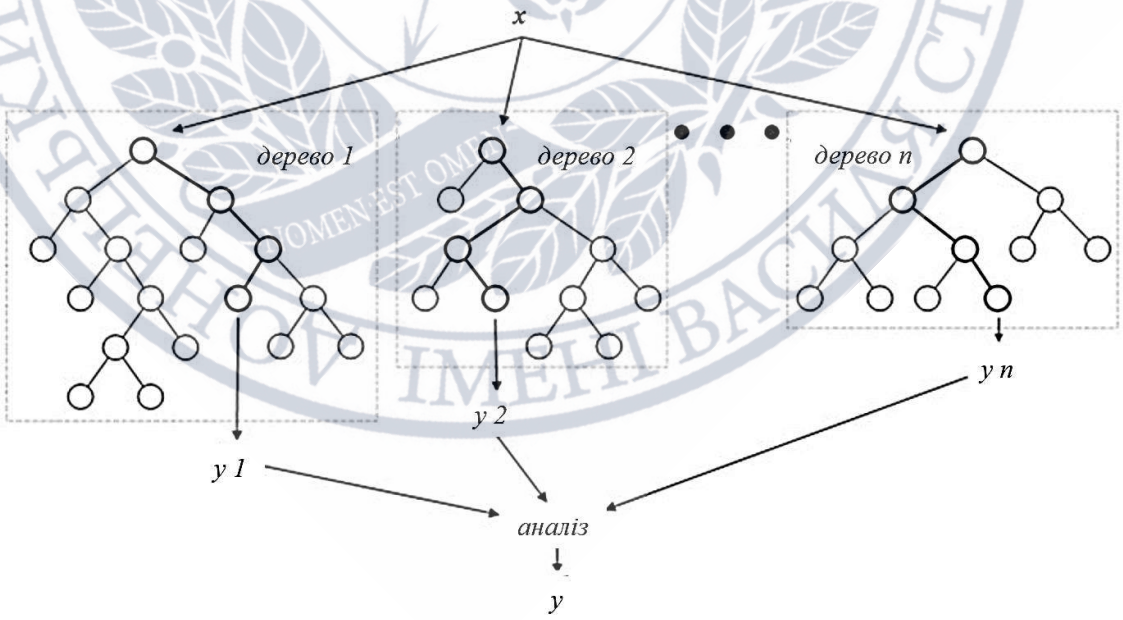

Рисунок 2.4. – Random Forest

Кожна гілка дерева веде до певного результату. В кінці алгоритм виводить дані, які є висновками відповідного дерева співвідносного атрибуту.

Далі алгоритм робить припущення на основі цієї інформації. Завдяки такій будові моделі можливість істнування похибки зменшується.

Недоліком цього алгоритму є невисока інтерпретація даних, потребує великих обчислювальних ресурсів та ймовірно високу затрату часу [33] [34].

Метод опорних векторів. Завдання алгоритму векторної машини підтримки – знайти гіперплощину у N-мірному просторі (N – кількість ознак), що чітко розподіляє точки даних. Йому потрібно знайти площу, яка має максимальну відстань між точками даних обох класів.

Гіперплощини – це визначений простір, який допомагає розподіляти точки даних. Вже самі точки даних, що розташовуються з різних сторін гіперплощини, можна віднести до різних категорій. Розмірність гіперплощини залежить від кількості ознак. Якщо кількість вхідних ознак дорівнює двом, то гіперплощиною є лише лінія. Якщо кількість вхідних властивостей дорівнює трьом, то гіперплощина стає двовимірною.

У методі опорних векторів беруться дані вихідної лінійної функції. Якщо вони більші за 1, тоді це категорія одного типу. Якщо вихід дорівнює -1, то даним присвоюються значення іншого класу. Оскільки порогові значення є 1 та -1, ми отримуємо цей діапазон значень [- 1,1], який виконує роль ось тієї площі, поля, що поділяє дані на різні категорії.

Недоліками цього алгоритму є те, що необхідно обраховувати ознаки для кожної категорії категорій. Існує також багато гіперпараметрів та їх значень, які не завжди можна розпізнати. Метод опорних векторів з труднощами обраховує великі набори даних [44].

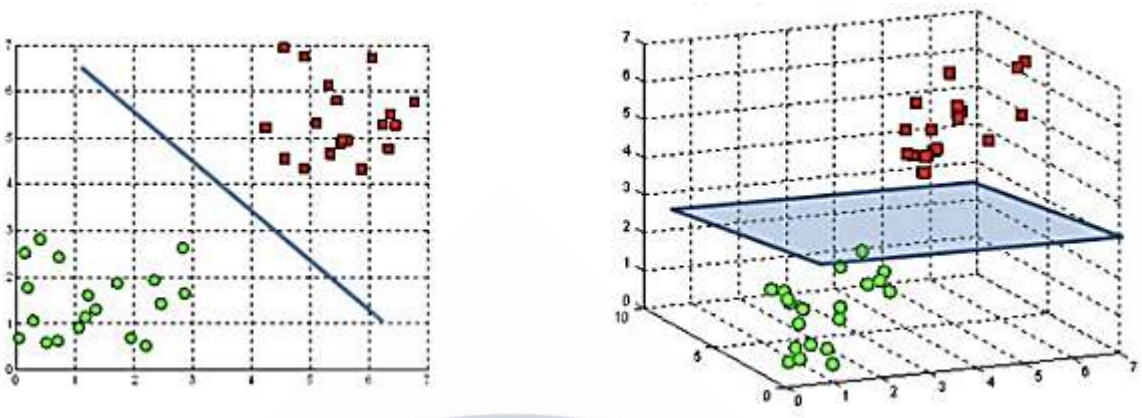

Рисунок 2.5. – Метод опорних векторів

• **Нейронні мережі** – це набір алгоритмів, розроблених вільно, подібно до роботи мозку людини. Вони інтерпретують особливості даних через своєрідне машинне сприйняття, маркування або групування вихідних даних. Шаблони, що розпізнаються, є чисельними, містяться у векторах, у які повинні бути переведені всі дані, будь то зображення, звук, текст чи часовий ряд. Нейронні мережі допомагають групувати немарковані дані за подібністю серед всієї вхідної інформації і класифікують їх, за розміченим набором даних для навчання.

Робота цього алгоритму має вигляд мережі, що складається з кількох рядів. Кожен ряд складається з вузлів. Вузол – це місце, де відбувається обчислення, схоже до нейрона в людському мозку, який спрацьовує, через вплив певної кількості подразників. Вузол поєднує входи даних з наборами коефіцієнтів або призначеною їх вагою, результатом чого є розгалуження, збільшення, або уніфікація, зменшення, даних. Таким чином утворюються відношення вхідних даних згідно із поставленим завданням. У цьому, наступному, створеному ряді підсумовується вага нових даних вузлів. Потім та сума передається через функцію активації вузла, щоб визначити, чи повинен і якою мірою цей сигнал просуватися далі по мережі, щоб дійти до кінцевого результату. Якщо сигнал пройшов далі – це означає, що нейрон був «активований» [33].

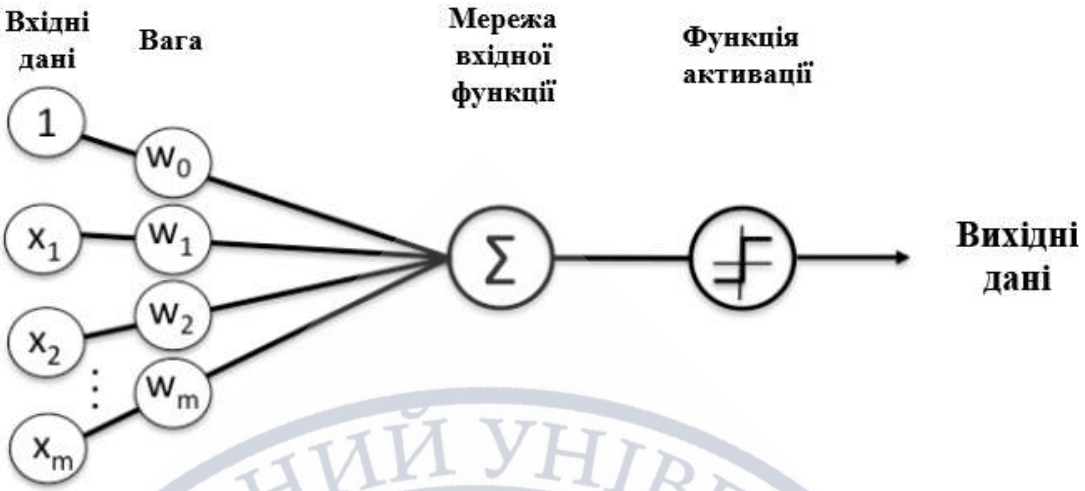

Рисунок 2.6. – Нейронна мережа

• **Наївний класифікатор Баєса** обчислює ймовірность того, що подія відбудеться, враховуючи, що інша подія вже відбулася. Для обчислення ймовірності того, що гіпотеза (h) є істинною, враховуючи попередні знання (d), теорема Байєса використовується так:

P (h | d) = (P (d | h) P (h)) / P (d),

де:

P (h | d) = Ймовірність про те, що відбудеться. Ймовірність того, що гіпотеза h є істинною, враховуючи дані d, де P (h | d) = P (d1 | h) P (d2 | h)…P (dn | h) P (d)

P (d | h) = Ймовірність. Ймовірність даних d з огляду на те, що гіпотеза h була істиною.

P (h) = Ймовірність того, що гіпотеза h є істинною (незалежно від даних)

P (d) = Ймовірність даних (незалежно від гіпотези).

Наприклад, потрібно обрахувати, якою буде сьогодні погода. Щоб визначити кінцевий результат – «так» або «ні», введемо змінну погоди – «сонячно» та обчислимо P (так | сонячно) і P (ні | сонячно), а потім оберемо результат з більшою ймовірністю.

P (так | сонячно) = (P (сонячно | так) * P (так)) / P (сонячно) = (3/9 * 9/14) / (5/14) = 0,60

P (ні | сонячно) = (P (сонячно | ні) * P (ні)) / P (сонячно) = (2/5 * 5/14) / (5/14) = 0.40

Отже, ймовірність того, що погода – «сонячна» є більшою, тому результатом буде відповідь – «так».

Основною ідеєю цього методу є те, що функції незалежні, що не завжди є правильно.

• **Метод k-найближчих сусідів.** Алгоритм k-найближчих сусідів використовує весь набір даних для навчання, а не розділяє його на навчальний та тестовий набори.

Щоб отримати результат нового екземпляру даних, алгоритм проходить через весь набір даних, щоб знайти k-найближчі екземпляри до нового, або k кількість екземплярів, найподібніших до нових. Потім алгоритм виводить середнє значення результатів. Значення k залежить від користувача.

Подібність екземплярів обчислюється за допомогою евклідової відстані (рис. 6) та відстані Хеммінга.

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + \ldots + (x_n - x'_n)^2}$$

Рисунок 2.7. – Евклідовий простір.

Недоліком цього алгоритму є низька ефективність для великих наборів даних, а також тривале та повільне обрахування значень [28].

– *неконтрольоване машинне навчання:*

• **Кластеризація методом k-середніх.** Під час опрацювання даних алгоритм починає пошук середніх значень, для кожного кластеру (об'єднання, групи, визначеної певними особливостями). Спочатку обраховуються випадково вибрані середні значення, які використовуються як початкові точки для кожного кластеру. Потім виконуються ітеративні (повторювані) обчислення для оптимізації центральних позицій. Алгоритм зупиняє створення та оптимізацію кластерів, якщо середні значення стабілізувалися, і жодні зміни їх значень не відбуваються, оскільки кластеризація пройшла успішно; або коли

визначену кількість ітерацій виконано. Отже, алгоритм він має лінійну складність O (n).

Перевага методу k-середніх в тому, що вона досить швидка, оскільки все, що він виконує, – це обчислення відстаней між точками та груповими центрами, де немає великої кількості обчислень. Проте цей алгоритм має кілька недоліків. По-перше, потрібно з'ясувати скільки груп / класів існує. Це не завжди просто і можливо. Метод k-середніх також починається з випадкового вибору центрів кластерів, і тому може давати різні результати кластеризації для різних запусків алгоритму. Таким чином, результати можуть бути не повторювані та не матимуть послідовності [17] [18].

- **T-розподілене вкладення стохастичної близькості** – це некерована нелінійна методика, що використовується в основному для дослідження даних та візуалізації даних багатовимірного простору. Спочатку алгоритм обчислює схожість між точками у просторі. Для кожної точки даних застосовуватиметься функція Гауса. Потім ці дані зводяться до певної норми. Цей крок дає набір ймовірностей для всіх точок. Далі використовуються використовуєте розподіл Стьюдента, що дасть другий набір ймовірностей для дво- чи тривимірного простору. Вкінці вимірюється різниця між розподілами ймовірностей двовимірних просторів, використовуючи відхилення Кульбака-Лейблера [21].

Рисунок 2.8. – Вимірювання парних подібностей у багатовимірному просторі

• **Метод головних компонентів** – це метод зменшення розмірності наборів даних, у яких зростає рівень інтерпретаційності, із мінімізацією втрат інформації. Це робиться шляхом створення нових некорельованих змінних, які послідовно максимізують дисперсію (міру розсіяння значень випадкової величини відносно середнього значення розподілу). Пошук таких нових змінних, основних компонентів, зводиться до вирішення проблеми власного значення / власного вектора, а нові змінні визначаються набором даних, а не апріорі. Отже, метод головних компонентів визначається як адаптивний метод аналізу даних. Він є адаптивним і в іншому розумінні, оскільки розроблені варіанти методики, пристосовані до різних типів даних і структур [41].

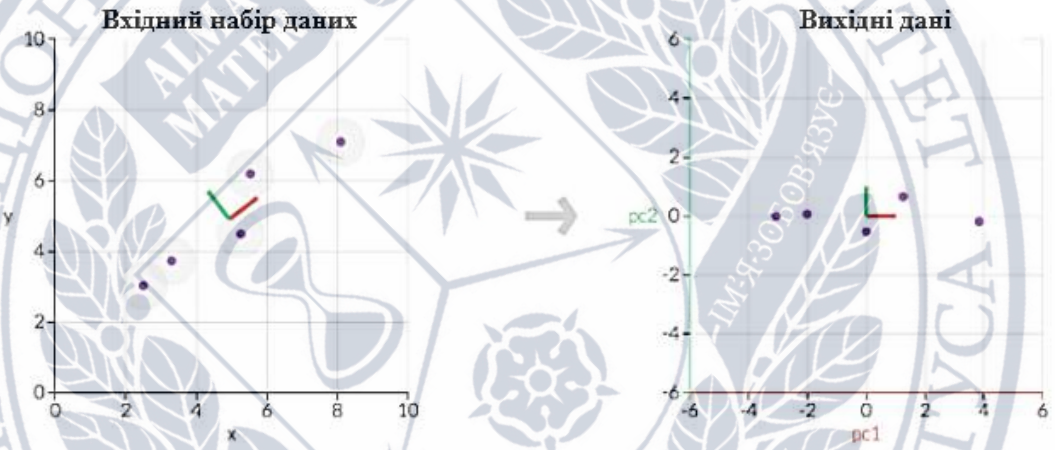

Рисунок 2.9. – Метод головних компонентів

• **Навчання на асоціативних правилах.** Цим алгоритмом користуються є сфері реклами, торгівлі та інших, де потрібен збір інформації, що ґрунтується на основі попередніх знань про властивість набору елементів. Застосувати цей алгоритм можна таким чином: потрібно задати набір послідовних операцій і правил, які будуть передбачати виникнення предмета на основі появи інших елементів для цього завдання. Отже, головним завданням є виявити зв'язки між змінними в наборі даних. Наприклад, правило {хліб, яйця} → {молоко}, знайдене в даних про продажі в супермаркеті, вказує на те, що якщо хтось купує суміш для хліб і яйця разом, то вони, ймовірно, також шукатимуть й молоко. Тоді набір продуктів буде таким: {хліб, яйця, молоко} [37].

– *навчання з підкріпленням:*

• **Q-навчання** – це алгоритм навчання, який прагне знайти найкращі рішення з урахуванням поточних даних. ‘Q’ в цьому методі означає якість. Якість у цьому випадку відображає, наскільки корисною є дія для отримання в майбутньому хорошого результату. Для цього потрібно створити q-таблицю або матрицю, яка відповідатиме формі, [стан, дія] та ініціалізувати значення до нуля. Ця таблиця стає довідковою для того, щоб алгоритм обрав найкращу дію на основі q-значення. Наступним кроком є взаємодія алгоритму з оточенням та оновлення пар станів дій у таблиці [стан, дія]. Q-навчання робить це двома способами. Перший – використовувати q-таблицю як орієнтир і переглянути всі можливі дії для заданого стану. Потім алгоритм обирає дію, виходячи з максимального їх значення. Цей спосіб називається експлуатацією, оскільки для прийняття рішення використовується вже наявна інформація. Другий спосіб полягає в тому, щоб діяти випадковим чином. Він називається дослідженням. Замість обґрунтованих рішень, алгоритм діє непродумано, випадково. Ця операція є важливою, оскільки дозволяє агенту досліджувати та виявляти нові стани, які можуть не бути обрані в процесі експлуатації. Встановивши, як часто робитиметься дослідження в порівнянні з експлуатацією, можна збалансувати їх використовуючи епсилон і встановивши значення того, як частоти застосування певних дій [44].

• **Temporal Difference (TD).** Це алгоритм, який навчається з інформації, яка надходить до нього через зразки даних без попереднього знання про цю конкретну інформацію. Це означає, що Temporal Difference використовує підхід до навчання без моделей або без учителя. Цей алгоритм можна назвати навчанням із спроб та помилок. В момент початку його роботи немає об’єктивної початкової оцінки даних, потрібно їх ініціалізувати, використовуючи випадкові значення або всі нулі, а вже потім вносити оновлення до цієї оцінки [45].

• **Monte-Carlo Tree Search (MCTS).** Monte-Carlo Tree Search передбачає навчання алгоритма так як і методу – Temporal Difference, взаємодіючи із вхідними даними та збираючи знання про них. Однак оцінка Monte-Carlo Tree

Search призначена лише для пробного навчання, тобто алгоритм може вдосконалювати власне навчання шляхом спроб і помилок, циклічно виконуючи свою роботу. У цьому процесі навчання кожна спроба називається епізодом, і всі епізоди повинні завершуватись, тобто слід досягти кінцевого результату. Значення кожного стану оновлюються лише базуючись на остаточній оцінці, а не на оцінках близьких до нього станів [36].

2.3.3. Засоби здійснення автоматичного синтаксичного аналізу

Усі ці засоби обробки інформації інтегруються в програми NLP. Для їх реалізації створено широкий набір бібліотек, що містять набори функцій, класів чи об'єктів для виконання необхідних завдань у процесі обробки природного мовлення. Саме на Python, високорівневій мові програмування, яка володіє простим синтаксисом, доступна велика кількість цих бібліотек.

Ось декілька основних бібліотек та архітектур, на прикладі яких можна здійснювати аналіз тексту:

1. **NLTK** – відкрита бібліотека широкого використання, створена 2001 року на базі мови програмування Python. За допомогою NLTK можна проводити статистичний аналіз тексту, синтаксичний і семантичний розбори та класифікацію тексту (рис. 1) [24].

```
In [38]: from nltk.tokenize import sent_tokenize, word_tokenize

In [42]: text1 = "Це місто було. Місто, яке було, за даними археологів, третиною тодішнього Києва. І воно тоді й виросло в цій долині,

In [43]: text1 = sent_tokenize(cs)
         text1

Out[43]: ['Це місто було.',
          'Місто, яке було, за даними археологів, третиною тодішнього Києва.',
          'І воно тоді й виросло в цій долині, в придністровській, на лівому березі завдяки його унікальному розташуванню.',
          'Це наявність води, по-друге, родючі землі, по-третє, захищеність від вітрів ну, і в крайньому разі, Дністер, та перетинан
         ня торгівельних шляхів: воно могло іти як по річці Дністер, так і впоперек річки.',
          'Були й піші, й водні шляхи.',
          'Моє прізвище Горбняк, вся моя родина жила на горбі, от тому ми і Горбняки.',
          'Село було прикордонним, на лівому березі Дністра.',
          'Це був 22 роки кордон з Румунією.',
          'Ви знаєте, то сьогодні нам смішно, то такі умови проживання, що сьогодні навіть уявити... Купатися не можна, ловити рибу
         теж, пасти корови не можна, у кожному селі — застава.',
          'Навпроти кожного села — мур в два метри.',
          'Співати не можна, свистати не можна, в яскравому ходити також.',
          'Перша постанова Ради Міністрів у Москві, яку підписав ще Косигін, звучала так: на Дністрі має бути збудовано каскад гідро
         електростанцій.',
          'Нам говорили про три, тільки ми не уявляли, що воно таке, бо тоді навіть що таке гідроекологічні, ніхто навіть взагалі не
         міг уявити, що це таке є.',
          '16 000 гектарів найродючіших земель, 100 гектарів лісу.']
```

Рис.2.10 Сегментація тексту

2. **SpaCy** – це безкоштовна бібліотека з відкритим вихідним кодом для розширеної обробки природної мови (NLP) на Python. Вона надає змогу вирішувати велику кількість лінгвістичних завдань. Деякі функції працюють незалежно від конкретної мовної моделі й можуть аналізувати текст відсутньої мови серед конвеєрів spaCy. Наприклад, це такі функції як виявлення меж речень і токенізація (рис.2).

Рис. 2.11 Токенізація

Інші функції (визначення частин мов і будування їхніх зв'язків залежності, лематизація, з'ясування подібностей лексем, розпізнавання іменованих сутностей) побудовані на статистичних даних. Вони працюють за навченими конвеєрами, наборами послідовних функцій, властивих тій чи тій мовній моделі. Якщо, наприклад, для італійської мовної моделі наявний компонент, що відповідає за морфологічні особливості лексем, то для румунської моделі його немає. Щоб отримати необхідний конвеєр функцій для мови, модель якої відсутня у списку серед доступних конвеєрів spaCy, потрібно провести її навчання. Для цього spaCy подає конфігурації, за якими можна це здійснити [39].

3. **FastText** – бібліотека, побудована на векторному представленні слів, тобто перетворенні лексеми в число. Вона використовується для віднайдення семантичної подібності та для класифікації тексту. FastText був опрацьований командою розробників Facebook. Зараз ця бібліотека пропонує попередньо навчені вектори слів для 157 мов [43], [44].

4. Бібліотеку **Gensim** розроблено для семантичної обробки тексту. Вона автоматично виявляє семантичну структуру документів, досліджуючи статистичні закономірності співпадіння в корпусі навчальних документів. Gensim побудована на алгоритмах неконтрольованого машинного навчання, тому вона може працювати із попередньо необробленими текстами [35].

5. **Трансформери** – це архітектура, яка за допомогою векторних розрахунків та власної системи кодування й декодування аналізує велику кількість вхідного тексту й знаходить в ньому залежності [25].

6. **BERT** – це архітектура машинного навчання з відкритим кодом для обробки природної мови (NLP). Її розроблено, щоб допомогти комп'ютерам зрозуміти контекст, використовуючи попередні та наступні тексти, тобто BERT – це трансформерна модель із подвійним декодуванним . Фреймворк BERT був попередньо навчений через бази текстів Вікіпедії. На своїх етапах дослідження система досягла новаторських результатів у питаннях розуміння природної мови, включаючи аналіз емоційно забарвленої лексики, маркування семантичних ролей, класифікацію речень та розшифровку багатозначних слів

Виконання цих завдань відрізняло BERT від попередніх мовних моделей, таких як word2vec і GloVe, які обмежені в аналізі контексту та багатозначних слів. BERT ефективно усуває неоднозначність, яка є найбільшою проблемою для розуміння природної мови [30].

BERT передбачає слово в пустому місці. Для цього моделі зазвичай потрібно тренувати, використовуючи велике сховище спеціалізованих позначених навчальних даних. Це вимагає трудомісткого ручного маркування даних командами лінгвістів. BERT використовує метод маскованого мовного моделювання, щоб мати фіксоване значення незалежно від його контексту. Потім BERT змушена ідентифікувати замасковане слово на основі лише контексту. У цій системі слова визначаються їх оточенням, а не попередньо фіксованою сутністю.

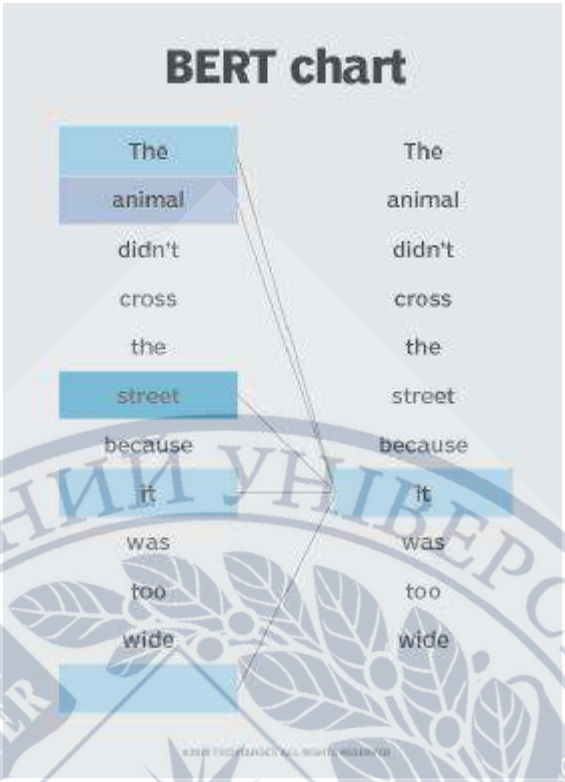

Рис. 2.12 Результат аналізу системи BERT

Архітектура BERT з часом змінювалась й вдосконалювалась. Зараз ми маємо кілька її версій, однією з яких є **ROBERTa**. RoBERTa має майже подібну архітектуру в порівнянні з BERT, але щоб покращити результати архітектури BERT, автори внесли деякі прості зміни в її архітектуру та процедуру навчання. RoBERTa не передбачає наступне речення, оскільки це трохи покращує продуктивність завдання. Додалась також можливість охоплювати більшу кількість даних з довшими послідовностями. Це має дві переваги: великі пакети покращують розуміння тексту мови, а також збільшують точність кінцевого завдання. В архітектурі BERT маскування виконується один раз під час попередньої обробки даних, в результаті чого утворюється єдина статична маска. Щоб уникнути використання єдиної статичної маски, навчальні дані дублюються і маскуються 10 разів, кожен раз з іншою стратегією маски через набір даних по 40 одиниць. Цю стратегію порівнюють з динамічним маскуванням [31].

7. **TensorFlow** – потужна бібліотека, призначена для машинного навчання. Її алгоритми дають змогу аналізувати також і великі обсяги тексту.

TensorFlow готує дані до навчання власних моделей – присвоює лексемам числові значення, щоб комп'ютер міг згодом розуміти ці слова й будувати нейронні мережі [42].

8. **Scikit-learn** – досить відома бібліотека машинного навчання. В NLP її застосовують для оцінки слів й будування векторів для розуміння та класифікації тексту [38].

Використання тих чи тих бібліотек залежить від досліджуваної мови та поставлених завдань. Якщо початкові етапи розпізнавання тексту (виявлення меж речень, токенізація) можуть виконувати багато бібліотек, то наступні кроки здійснюються спеціально призначеними алгоритмами (знаходження подібності слів, семантичний аналіз тексту і т.ін.) Загальна обробка мовлення є послідовною та складається з таких основних кроків:

1. **Сегментація тексту**.

Після завантаження тексту в систему машина бачить набір знаків і не має розуміння меж синтаксичних структур. Отже, для подальшого опрацювання тексту потрібно його поділити на менші одиниці – речення.

2. **Токенізація**.

Із наступним кроком система має поділити кожне речення на токени – усі елементи синтаксичної структури включно із пунктуаційними знаками, оскільки вони можуть стати важливою ланкою для деяких аспектів аналізу на наступних етапах.

3. **Тегування**.

Об'ємним та важливим етапом є присвоєння формальної інформації токенам – тегування за частиномовною належністю. Його базою є морфологічний аналіз елементів речення. Від результатів його роботи залежить успішність виконання синтаксичного аналізу.

Перед власне парсингом можна вдосконалити роботу алгоритмів обробки тексту з впровадженням проміжних етапів – лематизації, стемінгу та фільтрування стоп-слів.

Лематизація полягає у зведенні непрямих форм слова до початкової. Якщо машина правильно визначатиме базову форму, вона матиме можливість коректніше визначати зв'язки між елементами синтаксичної структури. Подібним процесом до лематизації є **стемінг**, який зводить лексеми до однієї основи. Відмінність роботи стемера від лематизатора полягає в охопленні аналізу слів з різною частиномовною належністю.

Фільтрування стоп-слів дає змогу позбутися токенів, що створюють «шум» і не є важливими в побудові синтаксичних залежностей (артиклі, допоміжні слова, дієслова-зв'язки, частки у творенні аналітичних форм слів). Після відкидання таких одиниць, залишаться слова, з якими можна проводити статистичний аналіз, пришвидшуючи процес аналізу на наступних етапах.

До роботи таких алгоритмів мають бути підключені словникові бази або списки слів, за якими і має проводитись відбір токенів. Проте на виході результатів існує вірогідність допущення помилок, пов'язаних з омонімією та поліфункційністю слів.

4. Парсинг.

Парсинг, або синтаксичний аналіз, досліджує синтаксичну залежність раніше проаналізованих токенів у межах поданих сегментів та будує зв'язки між ними. Більш вдосконалені алгоритми можуть визначати й типи зв'язку. Робота парсера базується на правилах граматики залежностей та статистичних даних (аналізі та тегуванні токенів, попередній обробці речень)

5. **Розпізнавання іменованих сутностей** (англ. Named entity recognition, NER).

Алгоритми NER працюють над розпізнаванням категорій імен у тексті (імена людей, місця проживання, назви організацій тощо). Ця техніка дає змогу проводити класифікацію та оптимізацію пошукових систем [18], [39].

Розпізнавання природного мовлення нині набирає обертів, оскільки полегшує виконання багатьох завдань. Велика кількість необробленого тексту, аудіо- та відеоматеріалу вже є розпізнана й багато такої інформації опрацьовується в реальному часі. Як бачимо, сучасні технології можуть

працювати із непідготовленим текстом, застосовуючи алгоритми глибокого навчання. Таким чином мови, для яких відсутні моделі для аналізу, можуть використовувати підготовлені конфігурації системами програмування, навчати власну модель, та проводити наступні дослідження.

РОЗДІЛ 3. АВТОМАТИЧНИЙ ЛІНГВІСТИЧНИЙ АНАЛІЗ

3.1. Автоматичний синтаксичний аналіз з допомогою бібліотеки Spacy.

У попередніх розділах було розглянуто моделі представлення речень, їхні типи, внутрішні зв'язки синтагм, а також способи адаптування цих моделей для автоматичного аналізу. За представленими алгоритмами та бібліотеками для природньої обробки мовлення ми проаналізуємо їх роботу для українських текстів.

Для дослідження було використано субтитри до відео з експедицій проекту «Ukraïner». Тексти проходили аналіз через алгоритми бібліотеки Spacy. Вибір цієї технології полягав у тому, що бібліотека Spacy поєднує в собі широкій набір функцій. Її конвеєр дає можливість визначати іменовані сутності та проводити морфологічний, синтаксичний аналізи, а також зв'язки між ними.

Оскільки конвеєр працює на лінгвістичних моделях, навчених системою, то для аналізу нашої збірки текстів, нам потрібно було обрати модель. Серед бібліотек Spacy дослідницької моделі для української мови не було, тому ми завантажили моделі для англійської та російської мов (рис. 3.1).

```
Ввод [4]: import spacy
          nlp = spacy.load("en_core_web_sm")
          nlp = spacy.load("ru_core_news_sm")
```

Рис. 3.1 – Завантаження моделей

Після завантаження даних ми зберегли документ у системі обробки для наступних кроків аналізу (3.2).

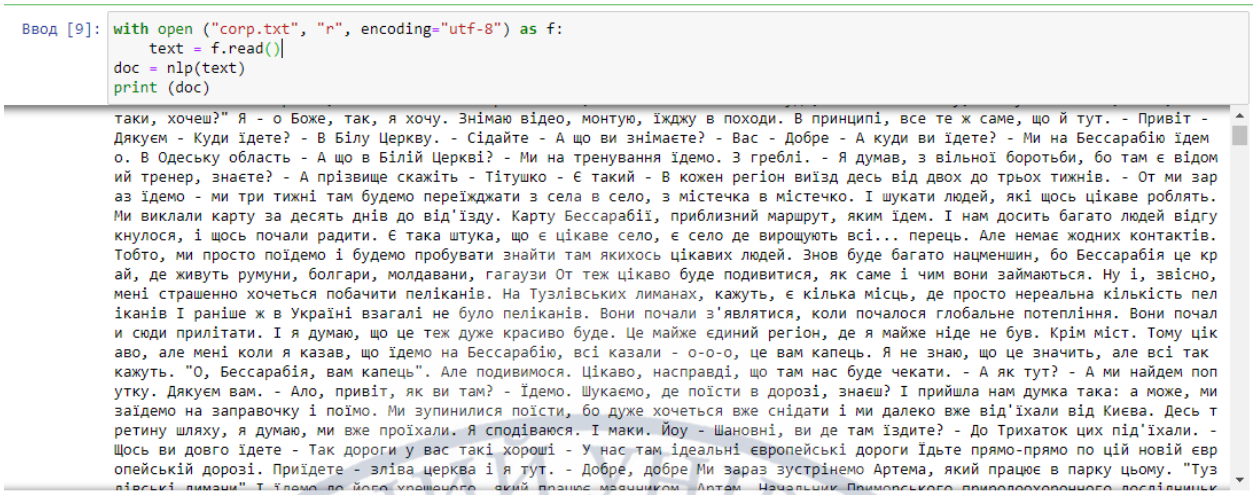

Рис. 3.2 – Створення doc для тексту

Наступними кроками були розподіл тексту на речення й токенізація (рис. 3.3, 3.4). За підрахунками всього токенів у документі 195 512 (рис. 3.5).

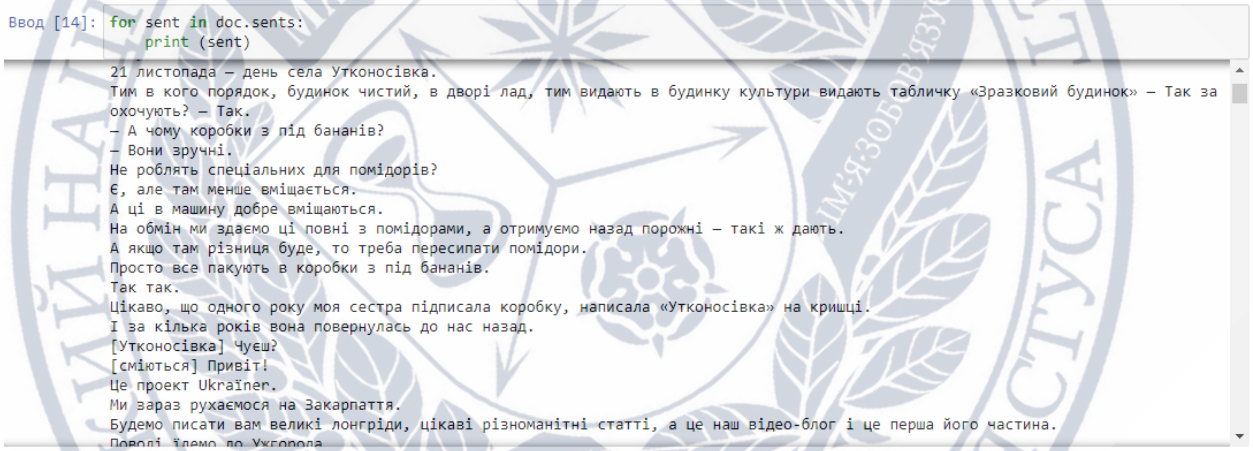

Рис. 3.3 Сегментація тексту

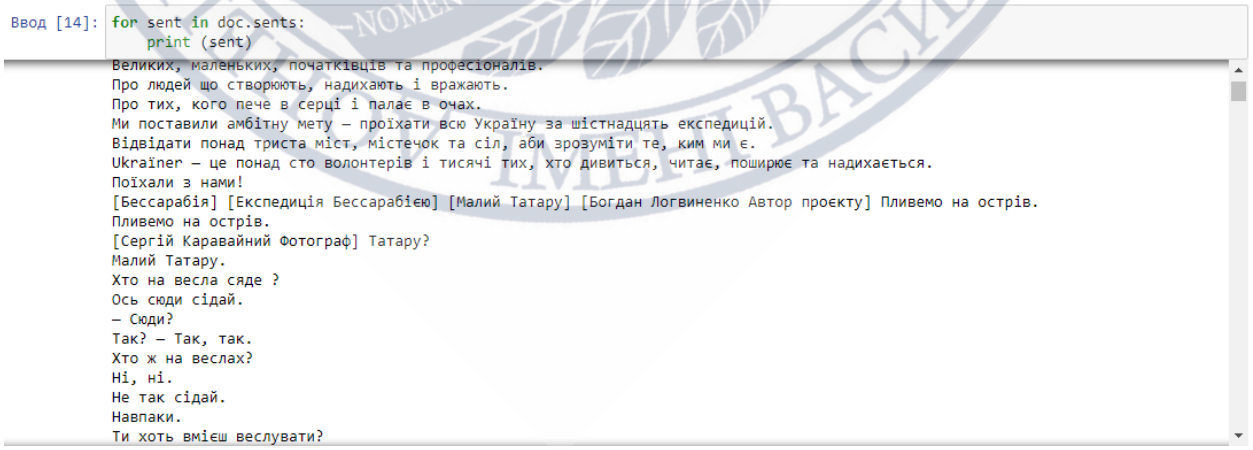

Рис. 3.4 Сегментація тексту

```
Ввод [111]: print (len(doc))

195512
```

Рис. 3.5

```
Ввод [59]: for token in doc[459:485]:
               print (token)

Бессарабії
,
приблизний
маршрут
,
яким
їдем
.
І
нам
досить
багато
людей
відгукнулося
,
і
щось
почали
радити
.
Є
така
штука
,
що
```

Рис. 3.6 Токенізація

```
Ввод [13]: for token in doc[:20]:
               print (token)

Ukrainer
-
це
історії
про
нас
,
українців
.
Великих
,
маленьких
,
початківців
і
професіоналів
Про
людей
,
що
```

Рис. 3.7 Токенізація

Spacy також визначає частиномовну приналежність – POS tagging (рис. 3.8, 3.9, 3.10, 3.11, 3.12). І для української мови алгоритми цього етапу конвеєра також працювали.

```
Ввод [71]: sentence = list(text_list[971])
           token = sentence[3]
           print(token)
           token.pos_

           важко

Out[71]: 'ADV'
```

Рис. 3.8 POS tagging

```
Ввод [61]: sentence = list(text_list[2389])
           token = sentence[3]
           print(token)
           token.pos_

           б

Out[61]: 'ADP'
```

Рис. 3.9 POS tagging

```
Ввод [57]: print(text_list[1244])

           І на пляшці виробник пише: "Спожити протягом трьох місяців" Що ж то за чудо ти сотворив, чоловіче добрий?

Ввод [58]: sentence = list(text_list[1244])
           token = sentence[7]
           print(token)
           token.pos_

           Спожити

Out[58]: 'VERB'
```

Рис. 3.10 POS tagging

```
Ввод [80]: print(text_list[4959])

           Центр сучасного мистецтва ще буде попереду.

Ввод [81]: sentence = list(text_list[4959])
           token = sentence[1]
           print(token)
           token.pos_

           сучасного

Out[81]: 'ADJ'
```

Рис. 3.11 POS tagging

```
Ввод [18]: print(text_list[40])

           І нам досить багато людей відгукнулося, і щось почали радити.

Ввод [55]: sentence = list(text_list[40])
           token = sentence[4]
           print(token)
           token.pos_

           людей

Out[55]: 'NOUN'
```

Рис. 3.12 POS tagging

Проте не всі слова були визначені правильно. На рисунку 3.13 і 3.14 можемо побачити, що система віднесла лексему «якусь» до дієслів, а «зрозуміли» - до прикметників. Spacy проводить аналіз подібності за допомогою системи векторів. Можна припустити, що відбулось сплутування опорних точок векторів, якщо система навчена модель не була раніше знайома з подібним контекстом.

```
Ввод [86]: print(text_list[4009])

           Як мінімум, отримуєш якусь інформацію.

Ввод [89]: sentence = list(text_list[4009])
           token = sentence[4]
           print(token)
           token.pos_

           якусь

Out[89]: 'VERB'
```

Рис. 3.13 Неправильне визначення частиномовної приналежності

```
Ввод [83]: print(text_list[4349])

           Він розуміє, що якщо тебе не зрозуміли в селі, не заплатили, дивляться на тебе дивно – значить туди вже не треба йти.

Ввод [85]: sentence = list(text_list[4349])
           token = sentence[7]
           print(token)
           token.pos_

           зрозуміли

Out[85]: 'ADJ'
```

Рис. 3.14 Неправильне визначення частиномовної приналежності

Останнім етапом аналізу був парсинг. Схеми синтаксичного аналізу можна побачити на наступних зображеннях (рис. 3.15, 3.16, 3.17):

```
In [46]: from spacy import displacy
         displacy.render(text_list[9], style="dep")
```

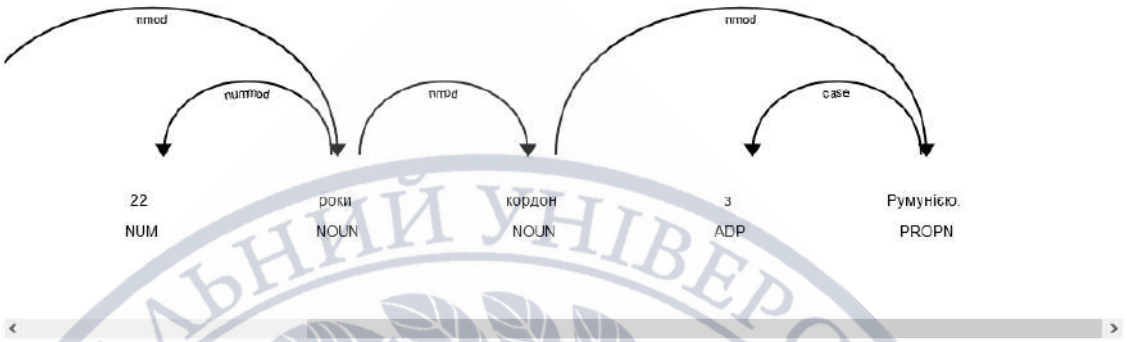

Рис. 3.15 Парсинг

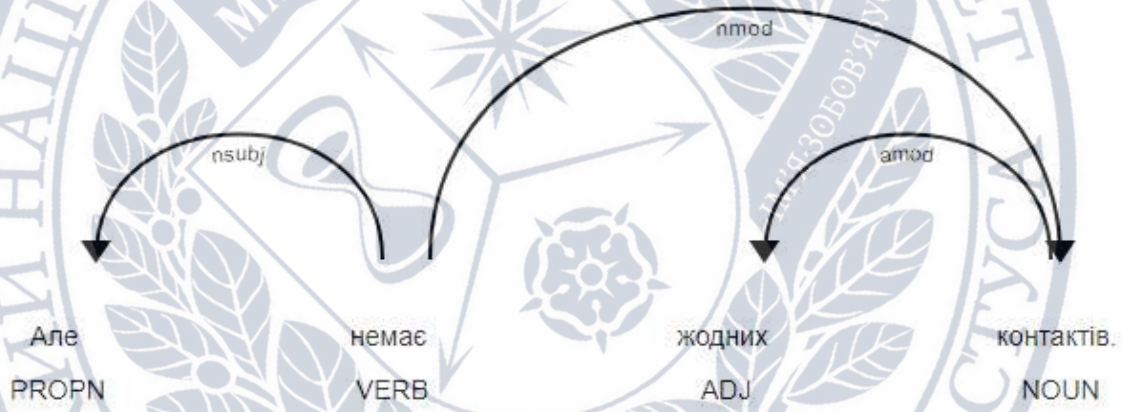

Рис. 3.16 Парсинг

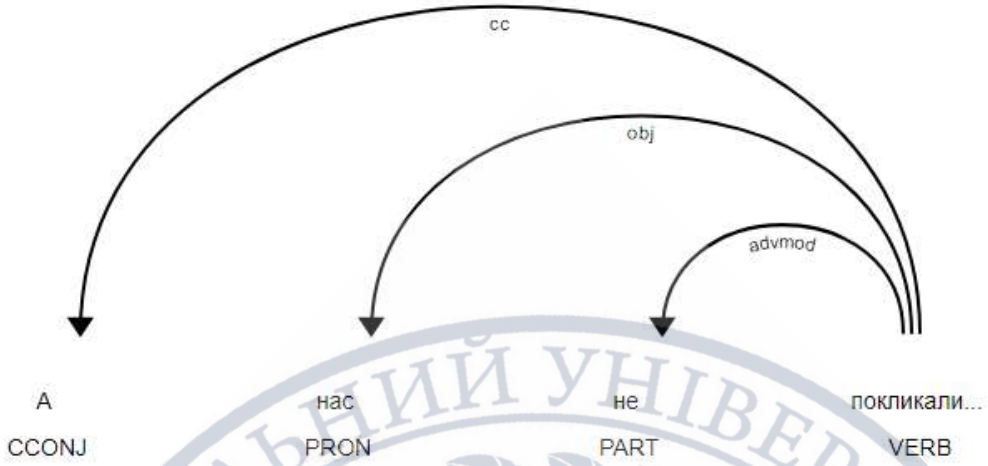

Рис. 3.17 Парсинг

Оскільки Spacy на попередньому етапі міг помилитися із морфологічним аналізом, то відповідно синтаксичний аналіз теж не завжди був точним. Приклади хибного аналізу наведені на рисунках 3.17 і 3.18, 3.19:

```
Ввод [97]: from spacy import displacy
           displacy.render(text_list[4232], style="dep")
```

Рис. 3.18 Хибний синтаксичний аналіз

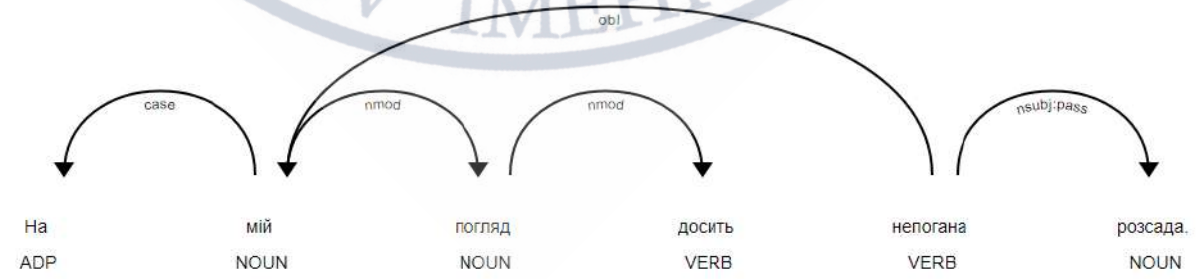

Рис. 3.19 Хибний синтаксичний аналіз

Можна зробити висновок, що алгоритми бібліотеки Spacy теж працюють для мов, що відсутні серед навчених лінгвістичних моделей, проте завантажувати потрібно споріднену їй модель.

Хто користується системою голосового введення Google, міг спостерігати, наскільки вищою стала їхня якість. Проте у ній відсутній модуль, який розпізнаватиме не лише мовлення, а й знаки пунктуації. Визначення місця розташування розділових знаків – одне з наступних завдань для повного автоматичного синтаксичного аналізу. Цю функцію можна реалізувати, якщо маємо спеціально навчену для цього модель. Алгоритми автоматичного визначення пунктуації вже існують, і вони доступні у відкритому доступі

Тому маємо завдання для майбутніх досліджень – навчання моделі для української мови та розроблення системи, яка автоматично буде розставляти розділові знаки.

ВИСНОВКИ

В результаті роботи було досягнуто мету – дослідити структуру української мови, а саме на морфологічному та синтаксичному рівні, для здійснення автоматичного аналізу в майбутньому. Основною матеріальною базою дослідження стали праці В. Г. Волошина, Т. А. Грязнухіної, Н. П. Дарчук, А. П. Загнітка, I. A. Bolshakov, D. Jurafsky, J. Martin. Як метод дослідження для опису мовної системи було використано моделювання.

Для проведення синтаксичного аналізу потрібно було встановити зв'язки між словоформами, а отже, виявлено необхідність визначити категорії цих словоформ, на чому й полягає морфологічний аналіз. З'ясовано, що для флективних мов існують певні труднощі, які полягають на парадигматизації. Щоб побудувати парадигму тієї чи тієї лексеми, слід виявити флексії та всі варіанти основи, а також визначити правила їх написання. Все це потребує остаточного зведення та переведення для здійснення автоматичного аналізу.

Після якісної морфологічної розмітки слідує синтаксичний аналіз. За матеріалами досліджень встановлено, що моделювання речення може відбуватися двома способами, а саме через граматики безпосередніх зв'язків або через граматики залежностей. Метод безпосередніх зв'язків полягає на бінарних відношеннях головного слова та залежного, які, утворюючи єдність, підпорядковуються чи підпорядковують іншу єдність, у такий спосіб в кінцевому підсумку зводяться до однієї цілісності – речення.

Ще один спосіб полягає в аналізі кожного слова окремо, де виділяється головне дієслово-присудок, якому підпорядковуються інші лексеми чи фразові структури. Однією з переваг такого методу є здатність обробляти мову з відносно вільним порядком слів, де відсутні правила побудови речення, а тому цей спосіб синтаксичного моделювання для української мови буде більш корисним.

Наступним етапом дослідження було з'ясувати методи та засоби автоматизації морфологічного та синтаксичного аналізу. Це можна зробити за

допомогою алгоритмів обробки природного мовлення, які базуються на статистичних даних та моделях машинного навчання й глибокого навчання.

Статистичні дані для обробки природного мовлення поєднуються у словники та корпуси текстів. Моделі машинного навчання, що можуть складатися з різних алгоритмів, які обираються залежно від поставленої мети, є різних ступенів автоматизації. Тобто проведення повторного аналізу буде залежати від попередньої перевірки вихідних даних людиною або ж самою програмою.

Для отримання результату спостереження та аналізу логіки допущення помилок доцільно було б використати такі алгоритми: навчання на асоціативних правилах, метод головних компонентів, Temporal Difference. Ці алгоритми не потребують великої кількості даних. Вони вдосконалюють своє навчання через спроби та проаналізовані раніше дані, хоч деякі з них на початковому етапі не мають конкретної їх оцінки.

Для синтаксичного аналізу можна також використати алгоритми методу навчання з учителем. Наприклад, нейронні мережі, оскільки робота цього алгоритму подібна до роботи мозку людини. Програма навчається за шаблонами, які дає їй людина. У нашому випадку це створені бази даних правильних та неправильних речень.

Виявлено, що саме високорівнева мова програмування Python має доступ до багатьох бібліотек, які можуть здійснювати автоматичний аналіз тексту. Деякі бібліотеки можуть проводити як морфологічний, так і синтаксичний аналіз. Багато тих використовують алгоритми побудови векторів, що визначають подібність слів та усувають неоднозначність. Досліджено також, що такі архітектури як RoBERTa можуть охоплювати велику кількість даних з довшими послідовностями, що може знадобитись для автоматичної пунктуації, яка зараз є мало досліджена для української мови через відсутність навчених моделей для автоматичного аналізу.

Серед етапів автоматичного аналізу природної мови було визначено такі: 1) сегментація тексту, 2) токенізація, 3) тегування, що полягає на проміжних

етапах – лематизації та стемінгу, 4) фільтрування стоп-слів, 5) парсинг, 6) розпізнавання іменованих сутностей, що вже стосується більш семантичного аналізу.

Здійснено аналіз роботи засобів розпізнавання текстової інформації для української мови, виходячи з доступних навчених моделей. Обробка мови відбувалась алгоритмами бібліотеки Spacy. Для аналізу було використано субтитри до відео з експедицій проекту «Ukraïner».

Завдяки широкому набору функцій Spacy та присутнім навченим моделям для англійської та російської мов ми мали змогу здійснити частковий синтаксичний аналіз текстів для української мови. Оскільки система не була знайома з усіма лексемами та словосполученнями, то алгоритм в результаті допускав помилки.

Отже, для повного автоматичного аналізу української мови, потрібно створити для неї навчену модель. Для цього корпус текстів має пройти через алгоритми навчання, що аналізуватимуть розмітку даних і будуть вводити дані в пам’ять системи. Пізніше таку модель можна буде використовувати для алгоритмів аналізу різних мовних рівнів, переведення мовлення в текст, машинного перекладу, автоматичного анотування та реферування текстів, що досі актуально для української мови. В перспективі ці алгоритми інтегруються до інших програмних забезпечень, що потребуватимуть лінгвістичного модулю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Базась М.Ф., Старовойт В.А., Скорина В.Є. Моделювання економічних процесів та прийняття управлінських рішень // Міжнародний збірник наукових праць. 2006. №3(6) С. 33–39.

2. Висоцька В. А. Особливості моделювання синтаксису речення слов'янських та германських мов за допомогою породжувальних контекстно-вільних граматик. В. А. Висоцька. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2015. № 246–276.

3. Волошин В.Г. Комп'ютерна лінгвістика: [навчальний посібник]. Суми: ВТД. "Університетська книга", 2004. 382 с.

4. Грязнухина Т. А. Синтаксический анализ научного текста на ЭВМ : Моногр.. Т. А. Грязнухина, Н. П. Дарчук, В. И. Критская, Н. П. Маловица, Т. К. Пуздырева; АН Украины. Ин-т языковедения им. А.А.Потебни. Київ : Наук. думка, 1999. 272 с.

5. Дарчук Н.П. Комп'ютерна лінгвістика (автоматичне опрацювання тексту): [підручник]. Київ: Видавничо-поліграфічний центр «Київський університет», 2008. 351 с.

6. Загнітко А. П. Теоретична граматика сучасної української мови. Морфологія. Синтаксис. Донецьк: ТОВ «ВКФ «БАО», 2011. 992 с.

7. Кульчицький І. М. Концептуалізація понять "модель" та "моделювання" у наукових дослідженнях / І. М. Кульчицький // Вісник Національного університету "Львівська політехніка". Серія : Інформаційні системи та мережі. 2015. № 829. С. 273-284.

8. Лангенбах М. Автоматичний синтаксичний аналіз речення за принципами граматики залежностей. М. Лангенбах. Науковий вісник Східноєвропейського національного університету імені Лесі Українки. Філологічні науки. Мовознавство. 2015. № 3. С. 249–254.

9. Лосев А. Ф. Введение в общую теорию языковых моделей / под ред. И. А. Василенко. 3-е. изд. Москва: Эдиториал УРСС, 2010. 296 с.

10. Лукач М. О. Типи лінгвістичних моделей та їх застосування для розв'язання лінгвістичних задач / М. О. Лукач // Вісник Національного університету "Львівська політехніка". Інформаційні системи та мережі. 2013. № 770. С. 143–153.

11. Мілян Н. Аналіз методів машинного навчання з вчителем / Н. Мілян. // Міжнародна студентська наук.-техн. конф. "Природничі та гуманітарні науки. Актуальні питання". Тернопіль: Тернопільський національний педагогічний університет імені В. Гнатюка, 2018. С. 51–52.

12. Тарануха В.Ю. Інтелектуальна обробка текстів: [навчальний посібник]. В. Ю. Тарануха. Київ: електронна публікація на сайті факультету, 2014. 80 с. (Режим доступу: http://www.csc.knu.ua/uk/library/books/taranukha-40.pdf (20.01.2020)).

13. Щербина Ю. М. Науковий напрям та навчальна дисципліна «Математична лінгвістика» / Ю.М. Щербина, В.А. Висоцька, Т.В. Шестакевич // Вісник Національного університету «Львівська політехніка» : Інформаційні системи та мережі. 2010. № 673. С. 374–-385.

14. Antworth E. Morphological Parsing with a Unification-Based Word Grammar / Evan L. Antworth. // SIL LANGUAGE TECHNOLOGY. https://software.sil.org/pc–kimmo/morphological–parsing/.

15. Bajaj P. Reinforcement learning [Електронний ресурс] / Prateek Bajaj – Режим доступу до ресурсу: https://www.geeksforgeeks.org/what-is-reinforcement-learning/ (20.12.2019).

16. Bolshakov I. A. Computational Linguistics: Models, Resources, Applications / I. A. Bolshakov, A. Gelbukh. – Mexico City: Centro de Investigacion en Computaci ´ on, Instituto Polit ´ ecnico Nacional, 2004. 186 с.

17. Dr. Michael. Understanding K-means Clustering in Machine Learning [Електронний ресурс] / Dr. Michael, J. Garbade // 2018 Режим доступу до

ресурсу: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 (12.08.2021).

18. Geitgey A. Natural Language Processing is Fun! [Електронний ресурс] / Adam Geitgey // Medium. 2018. Режим доступу до ресурсу: https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e (12.08.2021).

19. George S. The 5 Clustering Algorithms Data Scientists Need to Know [Електронний ресурс] / Seif George Режим доступу до ресурсу: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 (12.08.2021).

20. Gimon Z. Introduction to Machine Learning Algorithms for Beginners [Електронний ресурс] / Zee Gimon Режим доступу до ресурсу: https://huspi.com/blog-open/guide-to-machine-learning-algorithms (12.08.2021).

21. Gowthamy V. Machine Learning: Supervised Learning vs Unsupervised Learning [Електронний ресурс] / Vaseekaran Gowthamy. 2018. Режим доступу до ресурсу: https://medium.com/@gowthamy/machine-learning-supervised-learning-vs-unsupervised-learning-f1658e12a780 (20.01.2020).

22. Gupta A. ML | Semi-Supervised Learning [Електронний ресурс] / Alind Gupta Режим доступу до ресурсу: https://www.geeksforgeeks.org/ml-semi-supervised-learning/ (20.12.2019).

23. Gupta M. ML | Types of Learning – Supervised Learning [Електронний ресурс] / Mohit Gupta Режим доступу до ресурсу: https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/ (20.01.2020).

24. Jablonski J. Natural Language Processing With Python's NLTK Package [Електронний ресурс] / Joanna Jablonski // Real Python. 2021. Режим доступу до ресурсу: https://realpython.com/nltk-nlp-python/ (12.08.2021).

25. Joshi P. How do Transformers Work in NLP? A Guide to the Latest State-of-the-Art Models [Електронний ресурс] / Prateek Joshi // Analytics Vidhya. – 2019. Режим доступу до ресурсу:

https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/ (12.08.2021).

26. Jurafsky D. Speech and Langauge Processing / D. Jurafsky, J. Martin., 2000.

27. Kapoor A. Deep Learning vs Machine Learning: A Simple Explanation [Електронний ресурс] / Ajay Kapoor // Hacker Noon. 2020. Режим доступу до ресурсу: https://hackernoon.com/deep-learning-vs-machine-learning-a-simple-explanation-47405b3eef08 (12.08.2021).

28. KNN(K-Nearest Neighbour) algorithm, maths behind it and how to find the best value for K [Електронний ресурс]. 2019. Режим доступу до ресурсу: https://medium.com/@rdhawan201455/knn-k-nearest-neighbour-algorithm-maths-behind-it-and-how-to-find-the-best-value-for-k-6ff5b0955e3d (12.08.2021).

29. Kosmidou K. Models and modeling [Електронний ресурс] / K. Kosmidou, C. Zopounidis // Reference for business. Режим доступу до ресурсу: https://www.referenceforbusiness.com/management/Mar-No/Models-and-Modeling.html#ixzz6xaVY1Vuk (12.08.2021).

30. Lutkevich B. BERT language model [Електронний ресурс] / Ben Lutkevich // SearchEnterpriseAI. TechTarget. 2020. Режим доступу до ресурсу: https://searchenterpriseai.techtarget.com/definition/BERT-language-model (12.08.2021).

31. Overview of ROBERTa model [Електронний ресурс] // GeeksforGeeks. 2020. Режим доступу до ресурсу: https://www.geeksforgeeks.org/overview-of-roberta-model/ (12.08.2021).

32. Natural Language Processing (NLP) [Електронний ресурс] // IBM Cloud Education. 2020. Режим доступу до ресурсу: https://www.ibm.com/cloud/learn/natural-language-processing?mhsrc=ibmsearch_a&mhq=nlp (12.08.2021).

33. Nicholson C N. Some Basic Machine Learning Algorithms [Електронний ресурс] / Nicholson Chris Режим доступу до ресурсу: https://pathmind.com/wiki/machine-learning-algorithms (12.08.2021).

34. Reena S. The 10 Best Machine Learning Algorithms for Data Science Beginners [Електронний ресурс] / Shaw Reena Режим доступу до ресурсу: https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/ (12.08.2021).

35. Rehurek R. What is Gensim? [Електронний ресурс] / Radim Rehurek Режим доступу до ресурсу: https://radimrehurek.com/gensim/intro.html (12.08.2021).

36. Reinforcement Learning, Part 5: Monte-Carlo and Temporal-Difference Learning [Електронний ресурс]. 2019. Режим доступу до ресурсу: https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-5-monte-carlo-and-temporal-difference-learning-889053aba07d (12.08.2021).

37. Shaier S. ML Algorithms: One SD (σ)- Association Rule Learning Algorithms [Електронний ресурс] / Sagi Shaier // 2019 Режим доступу до ресурсу: https://medium.com/@Shaier/ml-algorithms-one-sd-%CF%83-association-rule-learning-algorithms-b35303e215d (12.08.2021).

38. Shaikh J. Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK [Електронний ресурс] / Javed Shaikh // Towards Data Science. 2017. Режим доступу до ресурсу: https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a (12.08.2021).

39. Shrivarsheni. Complete Guide to Natural Language Processing (NLP) with Practical Examples [Електронний ресурс] / Shrivarsheni // Machine Learning Plus. 2020. Режим доступу до ресурсу: https://www.machinelearningplus.com/nlp/natural-language-processing-guide/ (12.08.2021).

40. Shrivarsheni. spaCy Tutorial Complete Writeup [Електронний ресурс] / Shrivarsheni // Machine Learning Plus Режим доступу до ресурсу: https://www.machinelearningplus.com/spacy-tutorial-nlp/ (12.08.2021).

41. T. Jolliffe I. Principal component analysis: a review and recent developments / I. T. Jolliffe, J. Cadima. // The Royal Society. 2016. Режим доступу: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2015.0202 (12.06.2021).

42. Teja S. NLP with TensorFlow [Електронний ресурс] / Sai Teja // Medium. 2020. Режим доступу до ресурсу: https://medium.com/@saitejaponugoti/nlp-natural-language-processing-with-tensorflow-b2751aa8c460 (12.06.2021).

43. Text Classification & Word Representations using FastText (An NLP library by Facebook) [Електронний ресурс]. 2017. Режим доступу до ресурсу: https://www.analyticsvidhya.com/blog/2017/07/word-representations-text-classification-using-fasttext-nlp-facebook/ (12.06.2021).

44. Violante A. Simple Reinforcement Learning: Q-learning [Електронний ресурс] / Andre Violante // 2019. Режим доступу до ресурсу: https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56 (12.06.2021).

45. Violante A. Simple Reinforcement Learning: Temporal Difference Learning [Електронний ресурс] / Andre Violante // 2018. Режим доступу до ресурсу: https://medium.com/@violante.andre/simple-reinforcement-learning-temporal-difference-learning-e883ea0d65b0 (12.06.2021).

46. Word vectors for 157 languages [Електронний ресурс]. Режим доступу до ресурсу: https://fasttext.cc/docs/en/crawl-vectors.html (12.06.2021).