ШИФР «РУБІН»

КОНКУРСНА РОБОТА НА ТЕМУ:
«ТЕХНОЛОГІЇ СИНТАКСИЧНОГО АНАЛІЗАТОРА: МАШИННЕ НАВЧАННЯ»

Вінниця – 2020

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1 ТЕХНОЛОГІЇ АВТОМАТИЧНОГО СИНТАКСИЧНОГО АНАЛІЗУ 5

1.1. Лінгвістичний процесор та підходи автоматичного синтаксичного аналізу 5

1.2. Основні проблеми автоматичного синтаксичного аналізу 7

1.3. Етапи автоматичного синтаксичного аналізу 9

РОЗДІЛ 2 МАШИННЕ НАВЧАННЯ 12

2.1. Технології машинного навчання 12

2.2. Методи обробки природних мов 13

РОЗДІЛ 3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ СИНТАКСИЧНИХ СТРУКТУР 16

3.1. Концепція User Stories 16

3.2. Реалізація синтаксичного аналізу 17

ВИСНОВКИ 20

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 21

ВСТУП

Нині суспільство може спостерігати прогресивний розвиток та неймовірні досягнення сучасних технологій. Одним із актуальних, але найскладніших завдань є розроблення штучного інтелекту. Його опрацювання охоплює велику множину аспектів, з-поміж яких є оброблення природньої мови. Це завдання лінгвісти та програмісти намагаються вирішити ще із середини минулого століття. Її розв'язання передбачає створення корпусів текстів, систем автоматичного реферування й анотування, систем машинного перекладу, інформаційного пошуку та інші застосування, де є потреба мовного аналізу й синтезу.

Автоматичне розпізнавання текстів природної мови є одним із найпоширеніших завдань сучасних науковців. Вагомий внесок у перспективи вирішення цієї проблеми зробили Ноам Хомський (синтаксичні структури), Джон Бекус (синтаксис формальних мов), Джоакім Нівре (граматика залежностей), Даніель Юрафський, Джеймс Мартін (оброблення природньої мови). Серед вітчизняних науковців проблему автоматичного синтаксичного аналізу досліджували, а хто й дотепер над нею працює: В. Г. Волошин, Т. А. Грязнухина, Н. П. Дарчук, І. П. Білецька.

Ми маємо багато програмних продуктів, які не змогли б бути реалізованими без вбудованого в них лінгвістичного процесора. Наприклад, серія програмних додатків від Google – Google Translate, Google Voice Google Search та інші, які використовують словникові бази й алгоритми для перевірки та редагування правопису. Grammarly – додаток, за допомогою якого можна створити якісний текст англійською мовою, робити перевірки на різних мовних рівнях. MarketMuse – інструмент, який аналізує статті під час їх написання, даючи поради, щоб зміст був якнайбільш якісним. iPhone Siri – особистий помічник, що може спілкуватись із людиною та виконувати інструкції та прохання: зателефонувати другові, знайти ресторани та події, перевірити погоду та ін.

Мета дослідження полягає в розробленні програмного додатку, який здійснюватиме автоматизовану перевірку якості написання User Stories на синтаксичному рівні на базі бібліотеки NLTK.

Досягнення мети передбачає виконання низки завдань:

— розглянути технології автоматичного аналізу, а конкретно синтаксичного;

— дослідити роботу алгоритмів оброблення мовних структур;

— звернути увагу на застосування машинного навчання в межах цієї проблеми;

— створити базу речень User Stories, що містять слова, які найвірогідніше вживатимуться в роботі програми;

— написати код програмного додатку;

— розробити інтерфейс користувача.

Об’єкт дослідження – автоматичний синтаксичний аналіз.

Предмет дослідження – перевірка написання User Stories із застосуванням машинного навчання .

Методи дослідження. Під час дослідження в роботі були використані такі методи: аналіз, систематизація даних про автоматичний синтаксичний аналіз текстів, класифікування та опис матеріалу, синтез та узагальнення інформації, моделювання синтаксичних одиниць, діагностування частотної реалізації відповідних моделей.

РОЗДІЛ 1 ТЕХНОЛОГІЇ АВТОМАТИЧНОГО СИНТАКСИЧНОГО АНАЛІЗУ

1.1. Лінгвістичний процесор та підходи автоматичного синтаксичного аналізу

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 [10, 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 [12, с. 3-5]. Результатом машинного синтаксичного аналізу має стати автоматичне визначення структури елементів речення та зв'язків між ними [11, с. 134-135].

Існує два підходи до реалізації автоматичного синтаксичного аналізу – з використанням 1) **граматики безпосередніх зв'язків** та 2) **граматики залежностей**. Перший спосіб полягає в передаванні реченнєвої структури через бінарні зв'язки (головної частини та підпорядкованої):

{(Дівчинка) [грається (зі своїм котеням)]}.

Дане речення складається з таких елементів: іменних частин – *дівчинка*, *зі своїм котеням* та дієслівної частини – *грається зі своїм котеням*. Цю модель

можна зобразити також за допомогою синтаксичного дерева залежностей (рис. 1):

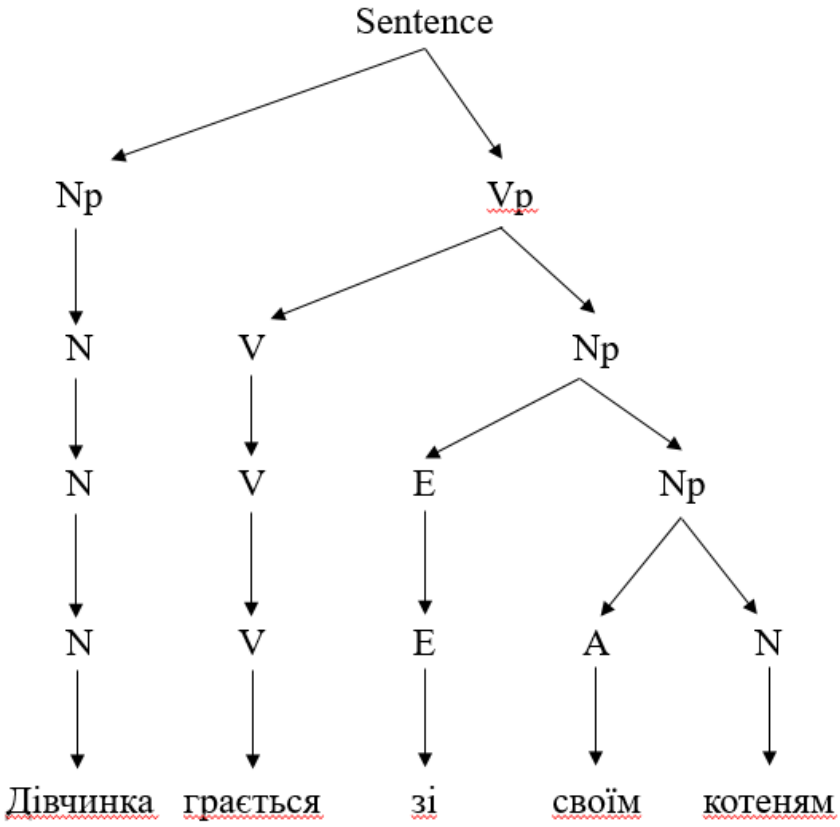

Рис. 1. Дерево безпосередньо складових

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 [11, с. 135-138; 6, с. 322-328], пор.: *Делегація молодих науковців приїжджає завтра*:

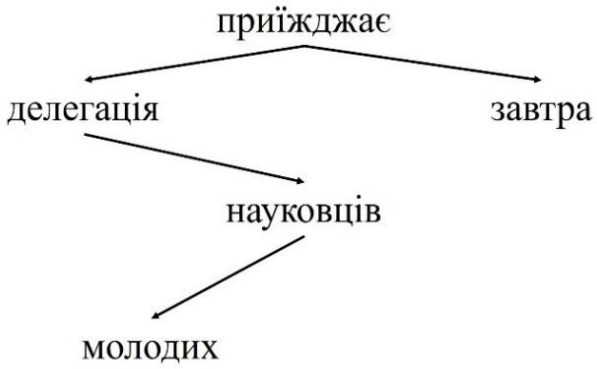

Рис. 2.Схема речення граматики залежностей

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.

1.2. 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**;*
- граматична конверсія. *«Розкопуйте **похованих** в 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;*
- 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* **(яка можливість? – прохання начальника уникнути; яке прохання? – уникнути начальника);**

• валентна варіативність – факультативність актантів, яка дає змогу відносити залежний елемент як до одного, так і до іншого потенційного головного слова: *Учитель співу не чує* **(учитель є глухим чи учитель не чує співу);**

• 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 до кількох 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 зовсім незрозумілою;

• порядок слів. Оскільки українській мові притаманний вільний порядок слів, то виникає велика кількість неоднозначних синтаксичних конструкцій: *Подейкували про можливий прихід військ Наливайка* [18].

1.3. Етапи автоматичного синтаксичного аналізу

Можна виділити універсальний алгоритм для автоматичного оброблення тексту. Тут виконуються такі кроки:

1. Сегментація тексту.

Після завантаження тексту в систему, машина бачить набір знаків і не має розуміння меж синтаксичних структур. Отже, для подальшого опрацювання тексту його потрібно 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 основою є морфологічний аналіз елементів речення. Від результатів його роботи залежить успішність виконання синтаксичного аналізу.

Для ефективного синтаксичного аналізу використовують парсинг, що ґрунтований на відповідному визначенні. Перед власне парсингом можна вдосконалити роботу алгоритмів оброблення тексту з впровадженням проміжних етапів – лематизації, стемінгу та фільтрування стоп-слів.

Лематизація полягає є в акумулюванні непрямих форм слова до початкової. Якщо машина правильно визначатиме базову форму, вона матиме можливість коректніше встановлювати 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 при творенні аналітичних форм слів). Після відкидання таких одиниць, залишаться слова, з якими можна проводити статистичний аналіз, пришвидшуючи процес аналізу на тих чи тих 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.

Наявне розуміння парсингу в інформатиці та комп'ютерній лінгвістиці. В інформатиці парсингом називають синтаксичний аналіз. У межах комп'ютерної лінгвістики парсинг розглядають як сукупність певних інструментів для досягнення відповідного результату в максимально короткий термін. Парсинг, або синтаксичний аналіз, досліджує синтаксичну залежність раніше проаналізованих токенів у межах поданих сегментів та вибудовує зв'язки між ними. Удосконаленіші алгоритми можуть визначати й типи зв'язку. Робота парсера ґрунтована на правилах граматики залежностей і статистичних даних (аналізі й тегуванні токенів, попередньому обробленні речень).

Додатковими етапами синтаксичного аналізу може стати розроблення алгоритмів на зняття омонімії, розпізнавання ключових слів, власних назв або ж інших однозначних слів, визначення зв’язку між синтаксичними сегментами за співвіднесеністю – різних частин мови та їхніх категорій. Ці операції не лише довершать автоматичний синтаксичний аналіз, а й постануть основою для подальшого семантичного аналізу [2] [10].

РОЗДІЛ 2 МАШИННЕ НАВЧАННЯ

2.1. Технології машинного навчання

Машинне навчання – це галузь науки про штучний інтелект. Його завдання полягає в тому, щоб змусити машину себе навчати, за допомогою початкових інструкцій та отриманого досвіду. Роботу програми за машинним навчанням можна назвати циклічною: отримання завдання – збір потрібних даних – побудова алгоритму – перевірка алгоритму – оброблення результатів – вдосконалення роботи за отриманими даними [9].

Синтаксичний аналіз на ґрунті машинного навчання працює за іншим принципом ніж класичні алгоритми. Для роботи цієї системи не потрібно прописувати всі етапи, зазначені вище, створювати спеціальні коди для кожного блоку в алгоритмі. Ідея машинного навчання полягає в тому, що існує загальний алгоритм, який оброблює вхідні дані.

Виділяють такі основні методи машинного навчання:

- 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) [5].

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 [16, 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 [3].

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 [4].

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 [1].

2.2. Методи обробки природних мов

Оброблення природних мов (NLP) – це автоматична або напівавтоматичне оброблення людської мови, а також загальний напрямок штучного інтелекту й математичної лінгвістики із застосуванням машинного навчання. NLP тісно пов’язана з мовознавством і має покликання на дослідження в когнітивній науці, психології, фізіології й математиці. Так, у сфері комп'ютерних наук NLP пов’язана з методами компілятора, формальною теорією мови, взаємозв’язком людини й комп’ютера, машинним навчанням і доказами теорем. NLP вивчає проблеми комп’ютерного аналізу і синтезу природних мов. Щодо штучного інтелекту, аналіз означає розуміння мови, а синтез – генерацію грамотного тексту. Вирішення цих проблем буде означати створення зручнішої форми

взаємодії комп’ютера і людини. До основних напрямів оброблення природної мови відносять такі, як витяг фактів, аналіз тональності тексту, відповіді на питання, інформаційний пошук, генерація тексту, переклад та ін.

Бібліотеки є одним зі способів оброблення природних мов. Серед них можна виділити такі: Stanford CoreNLP, NLTK, Scikit-learn, Tensorflow та інші. Кожна з вище названих бібліотек має свої переваги й недоліки. Детальніше про деякі з них:

- Stanford CoreNLP – цей безкоштовний програмний продукт був створений спільними зусиллями студентів і науковців університету Stanford. Основним завданням, яке було поставлене на початку його опрацювання, було створення набору сучасних інструментів, що дають змогу обробляти неструктурований текст. До цього дня цей продукт є одним із кращих у своїй ніші. З його допомогою можна провести повний аналіз частин мови в тексті, структури тексту, реалізувати розпізнавання іменованих об'єктів, визначити, де в тексті різні іменники позначають той самий об'єкт, провести аналіз тональності тексту й багато іншого.

- Natural Language Toolkit, NLTK – ця безкоштовна бібліотека для мови програмування Python є однією з кращих для створення різних програмних продуктів цією мовою. Вона надає великий набір інструментів, корпусів тексту, має передбачені обгортки для використання інших бібліотек всередині себе. Наприклад, для аналізу тональності тексту й розмітки пропозицій, є можливість підключення вищеописаного продукту Stanford CoreNLP. Також для різних класифікацій у NLTK був передбачений інтерфейс для підключень класифікаторів з іншої бібліотеки – Scikit learn.

- Scikit learn – ця бібліотека не має ніяких специфічних інструментів для оброблення природного мови, але в ній наявна величезна кількість класифікаторів, заснованих на різних алгоритмах; моделей нейронних мереж та інших спільних інструментів для машинного навчання. Використовуючи її разом з іншими, вузько направленими інструментами, можна створювати дуже складні

та якісні системи для оброблення, аналізу, класифікації й навіть генерації природного тексту.

- Tensorflow, створювалась не як бібліотека для оброблення природного тексту, а як інноваційна бібліотека для машинного навчання й особливо штучних нейронних мереж. Вона також може бути використана для аналізу й генерації природного тексту. Частиною аналізованої бібліотеки є seq2seq, і Word2Vec, які постійно допрацьовуються і вже успішно використовуються для різних досліджень і створення програмних продуктів.

РОЗДІЛ 3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ СИНТАКСИЧНИХ СТРУКТУР

3.1. Концепція User Stories

Для того, щоб звузити об’єкт нашого синтаксичного аналізу, ми обрали структуру речень User Stories.

User Stories – запити, що описують функційні можливості, які будуть цінними для користувача або покупця системи чи програмного забезпечення.

Між користувачем та виконавцем дії часто можуть виникнути непорозуміння. Різниця у знаннях різноманітних галузей є проблемою комунікації. Щоб домогтися успіху, у таких випадках користуються User Stories – твердженнями-запитами, які коротко описують потребу клієнта й дають змогу швидко зорієнтуватися виконавцю на вирішенні того чи того питання. Ще однією з переваг цих конструкцій є можливість зібрати велику кількість запитів, прохань, класифікувати їх та встановити пріоритетність їх виконання [7].

Найпоширеніший формат написання User Stories такий:

Як <суб’єкт/роль>, **я хочу** <ціль/потреба>, **для того щоб** <причина>.

Приклади:

1. Як розробник, я хочу замінити віджет папок, для того щоб у мене був найкращий віджет папок.

2. Як користувач, я хочу керувати рекламними оголошеннями, для того щоб видаляти застарілі або помилкові оголошення.

3. Як власник продукту, я хочу щоб в системі була можливість видаляти оголошення, для того щоб користувачі могли видаляти оголошення.

4. Як користувач, я хочу зберігати свої фотографії в системі, для того щоб мати можливість показати або продати їх іншим користувачам.

5. Як адміністратор, я хочу керувати фотографіями користувачів, для того щоб контент сайту був легальним.

6. Як веб-дизайнер, я хочу мати зв’язок з користувачами, для того щоб створювати вдалий дизайн сайту.

Отже, User Stories допомагають досягти належного етапу розуміння між людьми різних професій про те, що створювати, для кого, чому і коли. Оскільки їх легко визначити, зрозуміти й переглянути, вони можуть стати стандартним способом спілкування та узагальнення функційних можливостей продукту як для осіб з технічною освітою, так і без такої сукупності знань. Історії користувача усувають цей технічний вимір, тому будь-який член команди може зробити свій внесок просто як користувач.

Наше завдання полягає й тому, щоб переконатись, що користувач вноситиме правильні дані у форму та що User Stories будуть правильно написані з граматичного погляду.

3.2. Реалізація синтаксичного аналізу

У нашому дослідженні ми скористалися неконтрольованим машинним навчанням. Перед написанням програмного коду було створено дві бази речень User Stories. Одна база містить правильні User Stories – із адекватним дотримуванням їх структури, граматики й семантики. Другий масив речень складається із User Stories, що мають різні варіанти помилок. Наприклад:

1. *Як веб-дизайнера, я хочу отримав шаблона сторінка сайтів, для того щоб підібрати найкращий варіант для свого продукту.*
2. *Як веб-дизайнерах, я хочу отримаю шаблону сторінці сайтів, для того щоб підібрати найкращий варіант для свого продукту.*
3. *Як власникам контентом, я хочу мати можливість створювати контентів для продуктів, для того щоб я міг надавати інформацєю та пропонувати свою продукцію клієнтів.*
4. *Як власниках контент, я хочу мати можливість створювати контент для продуктів, для того щоб я міг надавати інформацію та пропонувати свою продукціями клієнтом.*

Спочатку система має проаналізувати ці дві бази. Після запуску програми користувачу надається інтерфейс, де він може ввести своє речення за структурою User Stories.

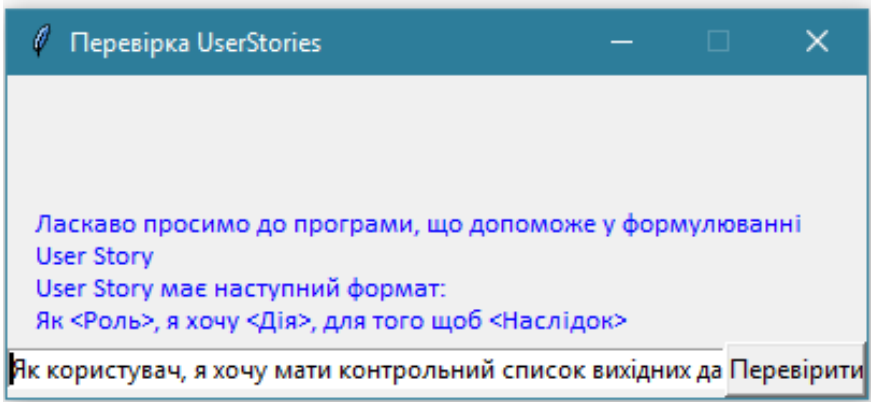

Рисунок 3.1. Інтерфейс користувача (із введеним реченням)

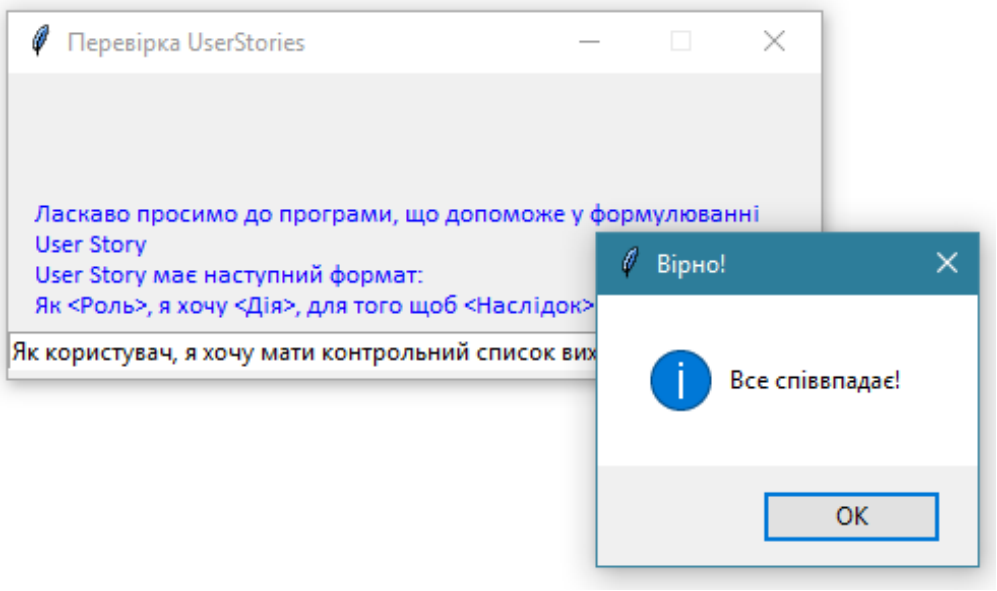

Рисунок 3.2 – Результат «Все співпадає!»

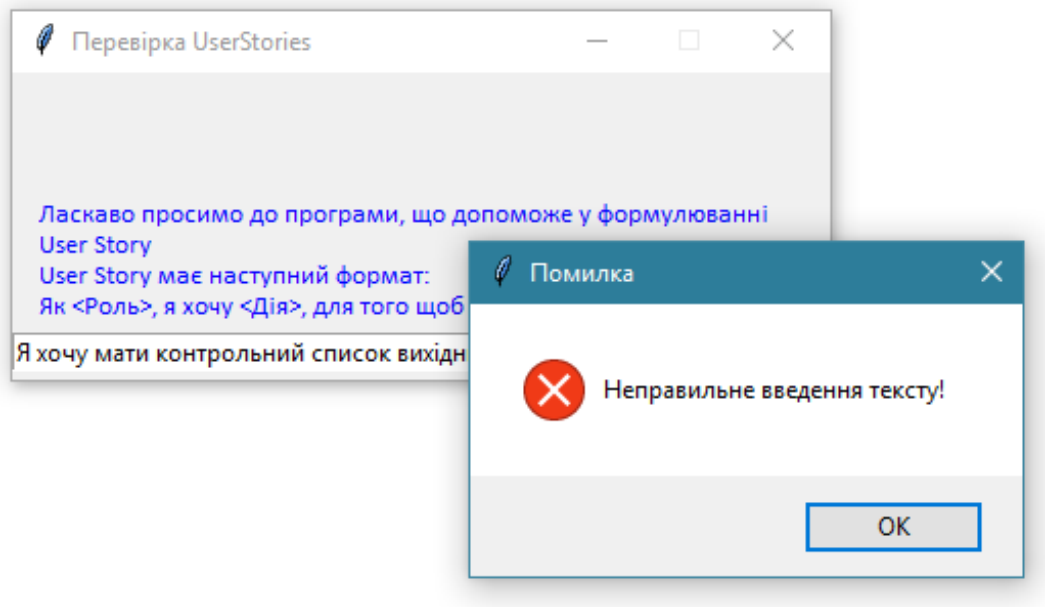

Рисунок 3.3 – Результат «Неправильне введення тексту!»

Програма знаходить помилки в реченнях. Але, на жаль, для досконалішого результату машині потрібна набагато більша база речень із врахуванням попередніх похибок у роботі програми, а отже застосувати напівавтоматичне машинне навчання.

ВИСНОВКИ

Згідно із результатами роботи можна зробити такі висновки:

1. Основними проблемами автоматичного синтаксичного аналізу є омонімія й полісемія, неоднозначність інтерпретації в синтаксичних конструкціях, валентна варіативність та виникнення синтаксичної неоднозначності.

2. Парсинг постає ключовим етапом у автоматичному синтаксичному аналізі, у якому генеруються зв’язки між словами і будуються дерева залежностей.

3. Навчання машин сьогодні охоплює опрацювання даних з участю людини, без її втручання, напівавтоматичне навчання та навчання на основі попередніх результатів роботи програми.

4. Нами було проаналізовано алгоритми автоматичного синтаксичного оброблення текстів. Ми взяли за основу автоматичного синтаксичного аналізу такі процеси: сегментацію, токенізацію, та парсинг. Встановлено, що парсинг є складним процесом, оскільки, щоб отримати якісний результат синтаксичного аналізу, потрібно виконати попередні етапи оброблення тексту й створити необхідні бази даних.

5. Створено базу речень User Stories, що містять слова, які найвірогідніше вживатимуться в роботі програми та послідовному алгоритмуванні правильності і/чи неправильності аналізованих речень за структурою, семантикою та ін..

6. Розроблено інтерфейс користувача, який виводить на екран результати перевірки написання User Stories.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

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

2. Geitgey A. Natural Language Processing is Fun! [Електронний ресурс]. Adam Geitgey – Режим доступу до ресурсу: https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e (20.12.2019).

3. 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).

4. 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).

5. 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).

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

7. Krasadakis G. How (and Why) to Write Great User Stories [Електронний ресурс]. George Krasadakis – Режим доступу до ресурсу: https://www.freecodecamp.org/news/how-and-why-to-write-great-user-stories-f5a110668246/ (20.12.2019).

8. Natural Language Toolkit [Електронний ресурс] – Режим доступу до ресурсу: https://www.nltk.org/ (20.12.2019).

9. Nesmiyanova A. What is Machine Learning and What is It Not? [Електронний ресурс] / Anna Nesmiyanova – Режим доступу до ресурсу: https://steelkiwi.com/blog/what-is-machine-learning/ (20.01.2020).

10. 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". Інформаційні системи та мережі. 2015. № 814. С. 246 – 276.

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

12. 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. К. : Наук. думка, 1999. 272 с.

13. Карпіловська Є.А. Вступ до прикладної лінгвістики: комп'ютерна лінгвістика: Підручник. – Донецьк: ТОВ «Юго-Восток, Лтд», 2006. – 188 с.

14. 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.

15. Митренина О. В. Проблемы неоднозначности синтаксического анализа : автореф. дис. на здобуття наук. ступеня канд. філ. наук : спец. 10.02.21 "Прикладная и математическая лингвистика диссертации на соискание ученой степени кандидата филологических наук". Санкт-Петербург, 2005.

16. 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.

17. Романюк, Юлія Віталіївна. Прикладна морфологія [Текст] : навч.-метод. посіб. : для студ. ден. та заоч. форм навчання напряму 0203 Гуманітарні науки 6.020303 - філологія освіт.-кваліфікац. рівня "бакалавр" / Ю. В. Романюк ; Черкас. нац. ун-т ім. Б. Хмельницького. – Черкаси : Вид. від. ЧНУ ім. Б. Хмельницького, 2010. – 120 с. – Бібліогр.: с. 117.

18. 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)).