Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____

Кафедра _____Програмної інженерії_____

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____другий (магістерський)_____

Дослідження методів якості розпізнавання мовлення

Виконав: студент 2 курсу, групи ІПЗм-17-1___
Спеціальності:
___121- Інженерія програмного забезпечення___
_____Дегтярьов Д.А._____

Керівник _____доц. каф. ПІ Турута О.П._____

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2019р.

Харківський національний університет радіоелектроніки

Факультет_Комп'ютерних наук_____

Кафедра_Програмної інженерії_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _121-Інженерія програмного забезпечення_____

освітньо-наукова програма _Інженерія програмного забезпечення_____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

«_____»_____ 20 ___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

Студентові *_Дегтярьову Дмитру Андрійовичу*_____

1. Тема роботи Дослідження методів якості розпізнавання мовлення
затверджена наказом по університету від "___" _____20 ___ р № _____
2. Термін подання студентом роботи до екзаменаційної комісії 11.06.2019р.
3. Вихідні дані до роботи (проекту) *аналіз та дослідження сучасних інтерфейсів та фреймворків що застосовуються при розробці інтерфейсу перетворення мови у текст та наступного аналізу та оцінки їх якості*
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) *_мета роботи, аналіз користувацьких і розробка функціональних вимог до програмної системи, аналіз існуючих рішень, мовна модель, аналіз сучасного стану розінавання мовлення, архітектура програмного забеспечення, опис програмної реалізації, тестування, причини помилок. Додатки: а) Апробація результатів роботи, б) слайди презентації, в)_електронні_матеріали_до_проекту_на_CD*_____
5. Перелік графічного матеріалу (назви слайдів презентації) *вступ, мета роботи, основні задачі роботи, актуальність проблеми, як все працює, системи розпізнавання мовлення, сучасний стан проблеми, підходи для вирішення проблеми, комп'ютерна лінгвістика, мовна модель , морвологічний аналіз, синтаксичний аналіз, новітні підходи, розуміння природной мови та її стадії, причини помилок, реалізация протопипу, діаграма системи, алгоритм роботм, демонстрація роботи додатку, результати роботи, висновки*
_____

6 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спец.частина	Турута О.П.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термин виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної галузі		
2	Огляд існуючих методів		
3	Методи аналізу та розпізнавання мовлення		
4	Підготовка пояснювальної записки		
5	Спецчастина		
6	Підготовка презентації та доповіді		
7	Попередній захист		
8	Нормоконтроль, рецензування		
9	Занесення диплома в електронний архів		
10	Допуск до захисту у зав. кафедри		

Дата видачі завдання _____ _____2019 р.

Студент _____

Керівник роботи _____Турута О.П

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до дипломної роботи: 105 сторінкок, 18 рисунків, 5 таблиць, 25 джерел інформації.

РОЗПІЗНАВАННЯ МОВИ, МОВЛЕННЯ, ЗВУК, СИНТАКСИЧНИЙ АНАЛІЗ, ЛЕКСИЧНИЙ АНАЛІЗ, АЛГОРИТМ, PYTHON, JAVA.

Метою роботи є аналіз сучасних інтерфейсів та фреймворків щодо застосування при розробці інтерфейсу перетворення мови у текст.

Значущістю дипломної роботи можна вважати те, що на даний момент в комп'ютерних технологіях задачі розпізнавання та розуміння контексту мови є дуже актуальними, тому що це може полегшити спосіб спілкування між людиною та комп'ютером, ці технології використовуються в медичних і військових застосуваннях, системах безпеки, автоматизованих системах розпізнавання та ідентифікації тощо.

Продовженням цієї роботи може бути реалізація програмного забезпечення з розширенням можливостей аналізу тексту, наприклад, реферування тексту, вилучення власних назв та дат, ввести поняття розмірностей, щоб програмне забезпечення не лише давало визначення, а також могло порівнювати предмети за різними характеристиками.

Результати даної роботи можуть бути використані в подальшому для поглибленого аналізу і реалізації складнішого програмного забезпечення.

SPEECH RECOGNITION, LINGUISTICS, SOUND, SYNTHESIS ANALYSIS, LEXICAL ANALYSIS, ALGORITHM, PYTHON, JAVA.

The aim of the thesis is the analysis of modern interfaces and frameworks for application in developing the interface of language conversion into text.

The significance of this work can be considered that at the moment in computer technologies the tasks of recognizing and understanding the context of speech are very relevant, since this can facilitate the way of communication between a person and a computer, these technologies are currently used in medical and military applications, security systems, automated systems for recognition and identification, etc.

The continuation of this work may be implementation of software with the expansion of the capabilities of neural networks, for example, the introduction of the concept of volume, so that the software does not only give definitions, but also could compare items by different characteristics.

The continuation of this work may be the implementation of software with the expansion of the ability to analyze the text, for example, referencing the text, extracting their own names and dates, to introduce the notion of dimensions, so that the software not only gave a definition, and also could compare the objects for different charac ics.

The results of this work can be used in the future for in-depth analysis and implementation of complex software.

ЗМІСТ

ВСТУП

Мова - це природний спосіб спілкування людей. Ми здобуваємо всі відповідні навички в ранньому дитинстві, покладаючись на мовні комунікації. Це відбувається настільки природно для нас, що ми не розуміємо, яким складним явищем є мова. Людський голосовий тракт і артикулятори - це біологічні органи з нелінійними властивостями, робота яких знаходиться не тільки під свідомим контролем, але також впливають різні фактори. В результаті вокалізація може змінюватися в широких межах з точки зору акценту, вимови, членороздільності, жорсткості, основного тону, обсягу і швидкості; крім того, під час передачі, наші неправильні мовні зразки можуть бути додатково спотворені фоновим шумом, а також електричними характеристиками (якщо використовуються телефони або інше електронне обладнання). Всі ці джерела мінливості роблять аналіз мови дуже складною проблемою.

Проте людям настільки комфортно з мовою, що ми також хотіли б взаємодіяти з нашими комп'ютерами через мову, замість того, щоб вдаватися до примітивних інтерфейсів, таких як клавіатура і вказівні пристрої. Інтерфейс мови буде підтримувати багато цінних додатків - наприклад, телефонний довідник, база даних мов виконання запитів для початківців, застосування в медицині або польових дослідженнях, автоматичний голосовий переклад на іноземні мови тощо. Такі спокусливі додатки мотивували дослідження в області автоматичного аналізу мови з 1950-х років. Великий прогрес був досягнутий з 1970-х, використовуючи ряд інженерних підходів, які містять шаблон узгодження, інженерні знання і статистичне моделювання. Проте, комп'ютери досі не наблизилися до рівня людини у розпізнаванні мови, і виявляється, що надалі значний прогрес потребує деяких нових ідей.

Що дозволяє людям так добре розпізнавати мову? Цікаво, що людський мозок працює під зовсім іншою обчислювальною парадигмою, ніж звичайний комп'ютер. Розвиток штучних нейронних мереж тісно пов'язаний з біологією. Штучний нейрон – це спрощена модель біологічного нейрона. Зв’язки

між нейронами, за аналогією зі зв’язками між природними нейронами, називаються синапсами. Штучний нейрон має єдиний вихід, який інколи називають аксоном. Штучні нейрони об’єднують, утворюючи при цьому штучні нейронні мережі.

Важливою властивістю нейронних мереж, що свідчить про їх великий потенціал і широкі прикладні можливості - паралельна обробка інформації великою кількістю нейронів. Завдяки цьому досягається значне пришвидшення обробки інформації. Іншою не менш важливою особливістю нейронних мереж є здатність до навчання та узагальнення інформації.

Останнім часом спостерігається тенденція зростання інтересу до використання нейронних мереж для вирішення різних завдань і застосування їх в різних сферах людського життя. З використанням нейронних мереж відкрилися можливості проведення обчислень у сферах, що до цього відносилися лише до сфери людського інтелекту. З’явилися можливості створення систем, які здатні вчитися, запам'ятовувати та аналізувати інформацію, що дуже нагадує розумові здібності людини.

Це все свідчить про те, що нейронні мережі можуть дійсно стати основою для системи розпізнавання та аналізу мови загального призначення, і що нейронні мережі пропонують деякі явні переваги в порівнянні з традиційними методами.

1 ОБ’ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

Метою даної роботи є аналіз та дослідження сучасних інтерфейсів та фреймворків що застосовуються при розробці інтерфейсу перетворення мови у текст та наступного аналізу.

Об’єкт дослідження - інтерфейси та фреймворки перетворення мови у текст, системи морфологічного та синтаксичного аналізу тексту.

Предмет дослідження - процес перетворення мови у текст, методи морфологічного, синтаксичного та лексичного аналізу тексту.

Досягнення основної мети наукового дослідження передбачає постановку і вирішення наступних взаємопов’язаних задач:

– дослідження процессу перетворення мови у текст;

– огляд існуючих інтерфейсів та фреймворків розпізнавання та аналізу мови;

– пошук та створення тестових даних для реалізації та проведення дослідів;

– визначення метрик та методів оцінювання систем аналізу тексту;

– розробка архітектури та реалізація застосування для тестування інтерфейсів та фреймворків.

Проаналізувавши існуючі публікації, можна зауважити, що існує багато дослідницьких статей, що висвічують тему розпізнавання та аналізу тексту. Однак сьогодні це все лише залишається актуальною проблемою, враховуючи те, що рівень перетворення мови у текст та методи аналізу не іноді показують незадовільний результат, також залишається багато мов, методи аналізу яких не були дослідженні. Саме тому в роботі відображено теоретичні відомості та приведено практичну реалізацію з результатами дослідів.

Робота буда побудована на основі роботи бакалаврату, а також аналізі та опрацюванню предметної області. Була підготовлена, а також опублікована доповідь-дослідження на тему “Survey of Speech Recognition Approaches”, вона була

представлена на молодіжному форумі “Радіоелектроніка та молодь в 21 столітті” в секції “Інформаційні інтелектуальні системи” (Додаток А).

На основі дослідження, аналізу та розробленного застосування можно робити наступні кроки на шляху до більш глибоко вивчення даної теми та реалізації складних програмних продуктів.

Перш за все, технології аналізу мови використовуються для голосового набору команд, в ситуаціях, при яких говорити набагато простіше, ніж друкувати. Розпізнавання мовлення застосовується в системах інтерактивного мовного самообслуговування, коли, на телефонні дзвінки в компанії відповідає робот, який може розібратися зі стандартними питаннями з області підтримки. Ще одне застосування технологій розпізнавання голосу - диктування текстів, своєрідний автоматичний секретар. Нарешті, все частіше з'являються системи з голосовим управлінням будь-якою технікою, наприклад «розумний дім», або автомобілем. Область застосування буде найближчим часом безперервно розширюватися у зв'язку з безсумнівною зручністю для користувача голосових команд та з прогресом в точності розпізнавання мови.

Коли точність розпізнавання мови підніметься до 95% -99%, всі будуть користуватися цією технологією. І різниця між 95% і 99% буде величезною. Ніхто не хоче чекати 10 секунд для відповіді. Точність, з подальшою затримкою - два ключові показники для системи аналізу мови.

Але навіть якщо не брати до уваги розпізнавання мови (ситуація з яким хоч і далека від ідеалу, але поліпшується з року в рік), то можна сказати, що в багатьох випадках заміна форм на єдине поле зі звичайним текстовим введенням допоможе зробити сервіс більш зручним та зрозумілим. Написав або сказав користувач, скажімо, «Два квитки до Харкова завтра вранці», і ваш сервіс тут же видав відповідні рейси! Або «У суботу о шостій футбол» - і подія збереглася в календарі! «Дмитро прийди завтра вранці на роботу раніше» - і потрібного контакту пішла sms, або призначити завдання в трекері завдань (а краще - і то і то). Тому і було вирішено протестувати інтерфейси та фреймворки розпізнавання мови на основі власного асистента, який вмів би розпізнавати голос та вносити дані до календаря.

2 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ

В останні роки основний тренд досліджень у сфері розпізнавання та аналізу мови зміщується в бік відмови від використання прихованих марковських моделей. Згідно марковським властивостям, наступний стан - в даному випадку звукова одиниця типу фонеми - в ланцюзі залежить тільки від попереднього стану і не залежить від всіх інших станів в минулому. Звичайно, така модель є дуже спрощеною, тому для побудови акустичних моделей в даний час стали використовуватися рекурентні нейронні мережі, які дозволяють зберегти довготривалі залежності.

Розвиток сучасних мовних технологій крокує в бік реалізації повного циклу навчання систем розпізнавання спонтанного мовлення без виділення окремих акустичних і лінгвістичних моделей. Замість попереднього відбору акустичних ознак всі ділянки мовного сигналу представляються своїми спектрограмами, які подаються на вхід однієї великої нейронної мережі. Далі ми більш детально зупинимося на майбутньому систем розпізнавання та аналізу мови, щоб окреслити актуальність питання, яке розглядається у роботі.

2.1 Аналіз предметної області

Світовий ринок інтелектуальних асистентів з 2012 року по 2014 рік зріс з $ 352 млн. до $ 572,2 млн.. До 2020 року очікується зростання ринку до $ 3,07 млрд, що становитиме 31% у порівнянні з ростом в 2013 році.

Поки одні компанії концентруються на створенні віртуальних помічників на веб-сторінках, інші приділяють увагу мобільним. На світовому ринку поки переважають великі компанії - творці віртуальних асистентів. На їх частку припадає 80% усієї виручки галузі. Прогнозовані області для збільшення зростання в цій сфері - транспортні, комунальні послуги, телекомунікаційний сектор.

Згідно зі звітом Transparency Market Research, найбільшою в світі виявилася частка північноамериканського ринку - 39%. З 2014 по 2022 рік, за прогнозами, найбільш швидкозростаючим стане азіатсько-тихоокеанський регіон - 33,4%.

Ринок тільки починає розвиток, тут ще не накопичено достатньо інформації і статистики для оцінки прибутковості, але потенціал у теми великий. При оптимістичному сценарії розвитку в найближчі 3-5 років ринок мовних технологій на території України може вирости до $ 100 млн в рік. У мовних технологіях і в смисловому аналізі текстів ключова роль залишиться за технологіями збору і обробки великих даних і технологіями побудови і навчання глибоких нейронних мереж.

Прибутковість даної галузі можна визначити, тільки якщо вирішити, що конкретно називати «ринком мовних технологій». Ринок програмного забезпечення для колл-центрів становить близько 2 млрд на рік, на технічну підтримку користувачів по телефону великі компанії витрачають десятки мільярдів на рік - так що оцінювати можна дуже по-різному.

Величезна кількість грошей отримують провайдери послуг телефонії для великих замовників. Банки, телекомунікаційні та страхові компанії мають сотні мільйонів користувачів, а це мільйони хвилин щодня. Будь-яка компанія, яка дозволяє автоматизувати роботу колл-центрів або великих замовників, має серйозні можливості для зростання.

Творець програми “Співбесідник HD” Андрій Єрмолаєв вважає, що інтелектуальні здібності помічників будуть ускладнюватися. Можливо, одного разу настане момент, коли користувачі в розмові не зможуть відрізнити чат-бота від людини. З іншого боку, отримують розвиток спеціалізовані додатки для покупки квитків і товарів, а також для отримання довідкової інформації.

Також Компанія Ovum прогнозує, що кількість різних пристроїв з голосовими помічниками в найближчі роки зросте як мінімум в два рази.

Мова йде про такі системи, як Amazon Alexa, Apple Siri, Google Assistant, Microsoft Cortana, Samsung Bixby тощо. Ці цифрові асистенти істотно спрощують отримання тієї чи іншої інформації, виконання повсякденних завдань, вирішення

питань і т. П.

За оцінками, в 2016 році по всьому світу використовувалося близько 3,5 млрд пристроїв з голосовими помічниками. При цьому основну масу таких гаджетів становили смартфони.

До 2021-го, як очікується, загальне число пристроїв з цифровими асистентами досягне 7,5 млрд. Причому ринок буде розвиватися не тільки за рахунок смартфонів і планшетів, але і завдяки впровадженню голосових помічників в побутові пристрої, а також в «розумні» автомобілі з інтернет-підключенням.

Аналітики вважають, що найбільшого поширення отримує асистент Google Assistant. Далі в рейтингу популярності розташуються китайські голосові помічники, Apple Siri і Samsung Bixby. Наступні позиції займуть асистенти Amazon Alexa і Microsoft Cortana.

В рамках щорічної конференції LSA 16 представник компанії-розробника інтелектуальних інтерфейсів MindMeld [1] Тімоті Татл заявив про те, що лише за останній рік використання голосового пошуку в загальній частці веб-пошуку зросла з 0 до 10%.

За даними Kleiner Perkins Caufield & Byers, більше 25% пошукових сесій користувачів в панелі Windows 10 здійснювалося за допомогою голосового взаємодії з інтерфейсом.

Настільки значне зростання популярності голосового пошуку можна пояснити помітним поліпшенням функціоналу персональних асистентів і швидким розвитком технологій.

2.2 Аналіз аналогів

Алгоритми та рішення таких гігантів, як Google, Apple будуть представленні та опрацьовані далі. Бо їх рішення достатньо повно покривають необхідні потреби

користувачів. Я знайшов менш популярні голосові асистенти з менш потужним функціоналом, так як моя система не може конкурувати з it гігантами. Хоча, звісно, лідерами ринку все ж є Amazon Echo, Google Now чи Google Assistant, Cortana, Siri.

«Дуся» - це голосовий асистент для Android, призначений для голосового управління смартфоном. У нього немає інтерфейсу - тільки маленька іконка в лівому верхньому кутку екрану. Додаток працює постійно у фоновому режимі і активується голосом, помахом руки, прикладанням смартфона до вуха або струшуванням і іншими способами.

Безліч можливостей для персоналізації дозволяє налаштувати асистент під себе. «Дуся» володіє значним словниковим запасом і ще більшим набором можливих сценаріїв для установки. Наприклад, скрипт для виклику таксі, для гри з додатком в міста, для перегляду онлайн-ТБ і пошуку музики. Спілкуватися на абстрактні теми «Дуся» не здатна - принаймні до установки відповідного чат-бот скрипта.

Як бачимо, функціонал достатній, але немає можливості вносити інформацію до календаря та і юзер інтерфейс не дуже привітний.

Наступний ассистент це - Speaktoit Цей голосовий помічник заснували вихідці Павло Сиротін, Артем Гончарук та Ілля Гельфенбейн. Після виходу він потрапив в топ-10 кращих програм для Android за версією The New York Times. Асистент доступний на Windows Phone, Android і iOS, помічника можна навчати новим фразам.

Як повідомляє The Wall Street Journal, стартап залучив $ 2,6 млн інвестицій в ході раунду B, який очолив фонд Motorola Solutions Venture Capital в липні 2014 року.

У майбутньому, за словами Іллі Гельфенбейна, віртуальні гаджети будуть дуже затребувані: «Ти даєш завдання, машина його виконує. Колись цей час має настати ». Кінцева мета розробників - «створити додаток, без якого неможливо жити», але потрібно на це близько 10 років. Гельфенбейн зазначив, що це перспективна галузь розвитку, оскільки носяться гаджетів стає більше, ніж комп'ютерів.

У найближчих планах Speaktoit - збільшити кількість підтримуваних мов і розширити діяльність. Компанія працює з великими автовиробниками і до 2020 року готує до випуску новий продукт. Головним ринком Гельфенбейн називає США. Можливо, тому, незважаючи на підтримку дев'яти мов, локалізація поступається в розпізнаванні мови іншим помічникам.

В іншому додаток мало відрізняється від конкурентів: підтримується набір номерів зі списку контактів і набір повідомлень, пошук в інтернеті, установка будильників і нагадувань. Тобто функціонал достатньо великий і покривають мої задачі. Інтерфейс приємний та інтуїтивний.

Хотів би закінчити аналіз наступним помічником «Співбесідника HD». Програму не раз відхиляли в App Store, посилаючись на повторення функціональності Siri. Через два місяці виправлений відповідно до рекомендацій додаток надійшло в App Store і майже відразу потрапило в топ-5.

Цей голосовий помічник називають аналогом Siri: його створила компанія-розробник iOS-додатків iDeveloper, не дочекавшись локалізації для iOS. Додаток практично не поступається західним зразком, а в чомусь навіть його перевершує. Наприклад, словниковий запас співрозмовника поповнюється призначеними для користувача питаннями, якщо вони не безглузді або нецензурні.

Решта функцій «Співбесідника HD» ті ж, що і у додатків-аналогів: встановлення будильника, пошук в інтернеті, набір повідомлень і дзвінки контактам з адресної книги. Одна з функцій асистента - гра в міста.

Як видно з порівняльного аналізу ринок насичений і є запит на персональних помічників з аналогічними функціями, авжеж моє рішення буде менш функціональне та більш вузько спрямоване.

2.3 Майбутнє систем розпізнавання мови

Машинний переклад (Machine Translation) [1] – найраніше додаток комп’ютерної лінгвістики, разом з яким виникла і розвивалася сама ця область.

Перші програми перекладу були побудовані в середині минулого століття і були засновані на найпростішої стратегії послівного перекладу. Однак досить швидко було усвідомлено, що машинний переклад вимагає набагато повнішої лінгвістичної моделі. В даний час існує цілий спектр комп'ютерних систем машинного перекладу (різної якості), від великих міжнародних дослідницьких проектів до комерційних автоматичних перекладачів. Істотний інтерес являють проекти багатомовного перекладу з використанням проміжного мови, на якій кодується сенс фраз, які перекладаються. Сучасний напрямок - статистична трансляція, яка спирається на статистику перекладних пар слів і словосполучень. Незважаючи на багато десятиліть досліджень цього завдання, якість машинного перекладу ще не досконала. Істотний прорив в цій області пов'язують з використанням машинного навчання і нейронних мереж.

Ще одне використання комп'ютерної лінгвістики - це інформаційний пошук (Information Retrieval) [2] і пов'язані з ним завдання індексування, реферування, класифікації документів.

Повнотекстовий пошук документів у великих базах текстових документів передбачає індексування текстів, що вимагає їх найпростішої лінгвістичної попередньої обробки, і створення спеціальних індексних структур. Відомо кілька моделей інформаційного пошуку, найбільш відома і використовувана - векторна модель, при якій інформаційний запит подається у вигляді набору слів, а відповідні (релевантні) документи визначаються на основі схожості запиту і вектора слів документа. Сучасні інтернет-пошуковики реалізують цю модель, виконуючи індексування текстів по вживаним в них словами і використовуючи для видачі релевантних документів витончені процедури ранжування. Актуальний напрямок досліджень в області інформаційного пошуку - багатомовний пошук по документам.

Реферування тексту (Summarization) - скорочення його обсягу і отримання короткого викладу його змісту - реферату, що робить більш швидким пошук в колекціях документів. Реферат може складатися також для кількох близьких по темі документів. Основним методом автоматичного реферування досі є відбір

найбільш значущих речень реферованих документів на основі статистики слів і словосполучень, а також структурних і лінгвістичних особливостей текстів.

Близьке до реферування завдання - анотування тексту документа та складання його анотації. У простій формі анотація являє собою перелік основних (ключових) тем у тексті, для виділення яких використовуються статистичні та лінгвістичні критерії.

При обробці великих колекцій документів актуальні завдання класифікації (Categorization) і кластеризації текстів (Text Clustering) [3]. Класифікація означає віднесення кожного документа до певного класу із заздалегідь відомими параметрами, а кластеризація - розбиття безлічі документів на підмножини тематично близьких документів. Для вирішення цих завдань застосовуються методи машинного навчання, в зв'язку з чим ці прикладні завдання часто відносять до напрямку Text Mining, оскільки він розглядався як частина наукової області Data Mining (інтелектуальний аналіз даних) [4]. Завдання класифікації набуває все більшого поширення, вона вирішується, наприклад при розпізнаванні спаму, класифікації SMS-повідомлень тощо.

Дуже близьке завдання до класифікації - рубріціорування тексту (Text Classification) - віднесення тексту до однієї з заздалегідь відомих тематичних рубрик (зазвичай рубрики утворюють ієрархічне дерево тематик).

Достатньо нове завдання, пов'язане з інформаційним пошуком - формувати відповіді на питання (Question Answering) [5]. Наприклад цікаве питання: «Хто зробив пароплан?». Проблема вирішується шляхом визначенням типу питання та пошуку текстів, які можуть містить відповідь на питання (звичайно використовують пошукові машини), і потім виділенням відповіді з знайдених текстів.

Актуальна прикладне завдання, часто яка відносять до напрямку Text Mining - це можливість вилучати інформацію з документу (Information Extraction) [6], що потребує вирішити задачу економічної та виробничої аналітики. Під час вирішення цього завдання здійснюється виділення певних об'єктів в тексті - іменованих сутностей (географічних назв, імен персоналій, назв фірм і т.д.), їх стосунків і

пов'язаних з ним подій. Вилучення інформації може зберігатися в таблицях бази даних або у фреймах і може бути згодом оброблена стандартними методами обробки структурованої інформації, наприклад, методами Data Mining. Під час вилучення цільової інформації ураховується лінгвістична інформація про текст, що отримується в результаті морфологічного, синтаксичного і, рідше, семантичного аналізу. У більшості випадків використовуються лінгвістичні ресурси - тезауруси, словники, а також знання про предметну область. Однією з під задач є дозвіл кореферентних зв'язків. Системи вилучення інформації з текстів предметно-залежні, тобто вимагають настройки на нові предметні області.

До напрямку Text Mining відносяться і два інші близькі завдання - виділення думок (Opinion Mining) і аналіз тональності текстів (Sentiment Analysis) [7], що привертають увагу все більшої кількості дослідників в силу своєї актуальності. У першому завданні відбувається пошук (в блогах, форумах, інтернет-магазинах та ін.) думок користувачів про товари та інші об'єкти, а також проводиться аналіз цих думок. Друге завдання близьке до класичного завдання контент-аналізу текстів масової комунікації, в ньому оцінюється загальна тональність висловлювань і документу вцілому. Найбільш відомі такі підходи до визначення тональності документу:

– заснований на словниках емотивної лексики і правилах, передбачає використання словників тональності для оцінки тональності окремих слів і всього документу. Правила використовуються для уточнення тональної оцінки висловлювань по тональності слів і їх контексту. Зазвичай виконується морфологічний і синтаксичний аналіз документу тексту;

– заснований на навчанні по розмічених вибірках текстів, передбачає автоматичне навчання за текстами, в яких вказані оцінки тональності і, можливо, об'єкти. В якості ознак навчання використовуються результати повного або часткового лінгвістичного аналізу;

– змішаний підхід (комбінація першого і другого підходів).

Зовсім інше прикладний напрямок, який розвивається повільно, але стійко - це автоматизація підготовки та редагування текстів на природних мовах. Одними з

перших досягнень в цьому напрямку були програми автоматичного визначення переносів слів і програми орфографічною перевірки документу(правопис, або автокорректор).

Ще однією прикладним завданням є навчання природним мовам, в рамках цього напрямку створюються комп'ютерні системи, що підтримують вивчення окремих аспектів (морфології, лексики, синтаксису) мови - англійської, російської та ін. Розробляються також багатофункціональні комп'ютерні словники, які не мають текстових аналогів і орієнтовані на широке коло користувачів, наприклад, словник сполучуваності слів російської мови КросЛексика [8], додатково надає довідки щодо синонімів, антонімів і інших зв'язкам слів за сенсом.

Наступний прикладний напрям, який варто згадати - це автоматична генерація текстів на природних мовах [9]. Це завдання можна вважати підзадачею вже розглянутої вище машинного перекладу, проте в рамках напрямку є ряд специфічних завдань. Таким завданням є багатомовна генерація, автоматична побудова відразу на декількох мовах спеціальних документів – патентних формул, інструкцій з експлуатації технічних виробів або програмних систем, виходячи з їх формальної специфікації.

3 КОМП’ЮТЕРНА ЛІНГВІСТИКА

Поява мережі Інтернет та бурхливе зростання доступної текстової інформації значно прискорило розвиток наукової галузі, яка існує вже багато десятків років і відомої як автоматична обробка текстів (Natural Language Processing) і комп'ютерна лінгвістика (Computational Linguistics). В рамках цієї області запропоновано багато перспективних ідей по автоматичній обробці текстів природною мовою (ПМ), які були втілені в багатьох прикладних системах, в тому числі комерційних.

3.1 Поняття комп’ютерної лінгвістики

Комп'ютерна лінгвістика - розділ науки, що вивчає застосування математичних моделей для опису лінгвістичних закономірностей. Її можна розділити на дві великі частини. Одна з них вивчає способи застосування обчислювальної техніки в лінгвістичних дослідженнях - застосування відомих математичних методів для виявлення закономірностей. Виявлені закономірності використовуються іншою частиною, що вивчає питання осмислення текстів, написаних природною мовою, - створення математичних моделей для розв'язання лінгвістичних задач і розробка програм, що функціонують на основі цих моделей.

Джерела комп’ютерної лінгвістики сходять до досліджень відомого американського лінгвіста Н. Хомського по формалізації структури природної мови [10], до перших експериментів з машинного перекладу, виконаними програмістами і математиками, а також до розробленими в області штучного інтелекту першими програмами розуміння природної мови [11].

Оскільки в комп’ютерній лінгвістиці об'єктом обробки виступають тексти природної мови, її розвиток неможливий без базових знань в області загальної

лінгвістики (мовознавства) [12]. Лінгвістика вивчає загальні закони природної мови - його структуру і функціонування, і включає такі області:

– фонологія - вивчає звуки мови і правила їх з'єднання при формуванні мови;

– морфологія - займається внутрішньою структурою і зовнішньою формою слів мови, включаючи частини мови і їх категорії;

– синтаксис - вивчає структуру речень, правила сполучуваності та порядку розташування слів у реченні, а також загальні його властивості як одиниці мови.

– семантика і прагматика - тісно пов'язані області: семантика займається змістом слів, речень та інших мовних одиниць, а прагматика - особливостями вираження цього сенсу у зв'язку з конкретними цілями спілкування;

– лексикографія описує лексикон конкретної ПМ - її окремі слова, їх граматичні і семантичні властивості, а також методи створення словників.

Трохи спрощене завдання, що є у комп'ютерної лінгвістики може бути сформульоване як розробка методів і засобів побудови лінгвістичних процесорів для різних прикладних завдань з автоматичної обробки текстів на природних мовах. Розробка лінгвістичного процесора для деякого прикладного завдання передбачає формальний опис лінгвістичних властивостей оброблюваного документу, яке можна розглядати як модель тексту (мови).

Основним поняттям лінгвістичного опису мови є поняття моделі мови. Довільна модель мови дозволяє формально описати мову, а точніше, ті з її аспектів, які необхідні для підвищення якості автоматичного розпізнавання мовлення. Описуючи можливу послідовність слів у фразі, ми піднімаємося на вищі рівні опису мовлення в порівнянні з фонетичним тому повинні враховувати системні відносини цих вищих порядків. Модель може бути складною , яка використовується для опису слова в реченні, що враховує синтаксичну чи семантичну структуру мовлення, чи може бути простою, коли поява будь-яких слів рівной мовірна (в такому випадку ми відмовляємося від лінгвістичного аналізу та обліку закономірностей і особливостей природної мови).

3.2 Мовна модель

Мовна модель - обов'язкова частина систем розпізнавання мовлення. Адже не будь-яка послідовність слів є реченням, між словами є граматичні та семантичні зв'язки. Мовна модель дозволяє дізнатися, які послідовності слів в мові більш вірогідні, а які менш.

Визначаючи можливу послідовність слів, ми піднімаємося на більш високі рівні опису мови в порівнянні з фонетичним і, як наслідок, повинні враховувати системні відносини вищих порядків. Використовувана модель опису слова в реченні може бути складною, що враховує синтаксичну та семантичну структуру висловлювання, а може бути дуже простою, яка вважає, що поява будь-яких слів рівноймовірна (в такому випадку ми відмовляємося від лінгвістичного аналізу та обліку закономірностей і особливостей природної мови).

Мовна модель - обов'язкова частина систем розпізнавання злитого мовлення. Не кожна послідовність слів є словосполученням (особливо для мов типу німецького - з жорстким порядком слів), між словами є граматичні і семантичні зв'язки. Мовна модель дозволяє дізнатися, які послідовності слів в мові більш вірогідні, а які менш.

Використання мовної моделі допомагає скоротити простір пошуку і зняти неоднозначність при виборі з декількох близьких за вартістю акустичних гіпотез (для української мови, наприклад, допомагає правильно розпізнати слово в потрібному відмінку).

Загальноприйнятим критерієм оцінки моделей мови у відриві від акустичної моделі є перплексія (або коефіцієнт невизначеності - perplexity), який відповідає середньому коефіцієнту розгалуження після кожного слова, відповідно до моделі мови. Перплексія - міра здатності моделі передбачати невідомі та нові послідовності слів для системи.

Чим нижче перплексія, тим краще модель мови. Але прямої залежності між зменшенням перплексії та поліпшенням якості розпізнавання мовлення немає,

зменшення перплексії більш, ніж на 10% зазвичай відбувається на якості розпізнавання. Очевидно, що для моделі мови, де всі слова рівно ймовірні і вірогідність появи слів не може залежати від оточення, то перплексія дорівнює розміру словника. У міру обліку залежностей між словами, перплексія зменшується до певної межі.

Загальна схема обробки текстів (рисунок 3.1) інваріантна стосовно вибору природної мови. Незалежно від того, якою мовою написаний вихідний текст, його аналіз проходить одні й ті ж стадії. Перші дві стадії (розбиття тексту або документу на окремі речення та на слова) практично однакові для більшості природних мов. Специфічні для вибраної мови риси можуть проявитися на етапі обробки скорочень слів і обробки розділових знаків.

Семантичний аналіз грунтується на результатах роботи попередніх фаз обробки документу тексту, які завжди специфічні для конкретної мови. Отже, способи подання їх результатів теж можуть сильно варіюватися, справляючи великий вплив на реалізацію методів семантичного аналізу. Результати аналізу, проведеного на ранніх стадіях, можуть бути багатозначні: для вихідних параметрів вказується не одне, а відразу кілька можливих значень (кілька значень слова). Тоді наступні стадії повинні вибирати більш ймовірні значення з результатів ранніх стадій аналізу і вже на їх основі проводити наступний аналіз.

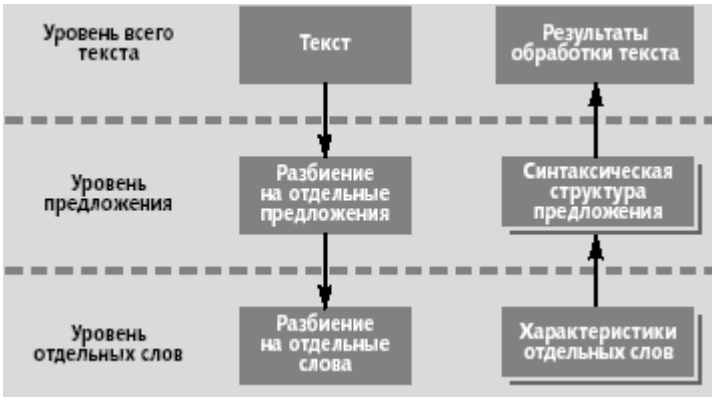

Рисунок 3.1 - Загальна схема обробки тексту

Розглянемо докладніше кожну із стадій аналізу документу після поділу тексту на окремі слова і фрази. До першої стадії (аналіз окремих слів) відноситься морфологічний аналіз (визначення частини мови, відмінка, дієвідміни тощо) і

морфемний аналіз (префікс, корінь, суфікс і закінчення); до другої стадії - синтаксичний аналіз; до третьої - різні завдання семантичного аналізу (пошук фрагментів, формалізація, реферування тощо).

Семантику, зміст, в мові пов'язують насамперед з поняттям значення (meaning). Значення слова - це його тлумачення, що приводиться в тлумачному словнику. Для відображення відносин між словами створюються спеціальні лінгвістичні ресурси - тезауруси, в яких слова зазвичай пов'язані відносинами синонімії, гіпонімії, або будь-яким асоціативним зв'язком. Найбільш відомий тезаурус WordNet містить більше 150 тис. слів і понад 170 тис. синонімічних рядів.

Відомо, що хороший статистичний опис мови в цілому вимагає наявності не тільки дуже великого тренувального корпусу, а й максимально широкого охоплення текстів різних жанрів і стилів. З іншого боку, якщо для певного тестового тексту ми будемо знати його приналежність до якого-небудь класу (наприклад, економіка, юриспруденція тощо), то використання для такого тексту моделі, натренованої на текстах відповідного класу, призведе до серйозного поліпшення якості розпізнавання. По суті, ми використовуємо різні моделі мови для текстів з різної темою. Основа моделі залишається однією і тією ж, але розподіл ймовірностей виявляється різним.

4 СИСТЕМИ РОЗПІЗНАВАННЯ МОВИ

Дикторозалежна система призначена для використання одним диктором, в той час як дикторонезалежна система призначена для роботи з будь-яким диктором. Дикторо-незалежність - дуже цінна якість системи, але в той самий час її дуже важко досягти, так як при навчанні системи вона налаштовується на параметри того диктора, на прикладі якого навчається. Таким чином, в процесі створення дикторонезалежної системи застосовуються набагато більш складні алгоритми навчання. У таких системах частота помилок розпізнавання зазвичай в 3-5 разів більше, ніж в дикторозалежних.

4.1 Характеристики систем розпізнавання мови

На даний момент системи розпізнавання та аналізу мовлення характеризуються такими ознаками:

– роздільність вимови;

– дикторо-залежність;

– призначення.

Якщо в мові всі слова розділяються інтервалами тиші пустотами, то ця мова - роздільна. А ось природна мова - злита. Аналіз злитої фрази набагато важчий, бо межі окремих слів не чітко визначені і тому їх вимова спотворена злиттям вимовлених звуків.

Призначення системи визначає необхідний рівень абстракції, з яким буде відбуватися розпізнавання та аналіз мови. Можна виділити два типи систем:

– командні;

– системи диктування.

У командних системах аналіз мовлення відбувається як аналіз цільного мовного елемента. Тобто, при розпізнаванні враховуються тільки фізичні характеристики сигналу, а не змістове навантаження промови.

Системи диктування також аналізують контекст мовного елемента і тому вимагають більшої точності розпізнавання. Алгоритми, залучені в таких системах, наприклад, приховані мережі Маркова, аналізують не тільки унікальні параметри самого мовного сигналу, але і контекст кожного вимовленого мовного елемента. Також можуть застосовуватися алгоритми динамічного програмування. Для аналізу контексту в системі необхідно передбачити набір граматичних правил, яким повинен задовольняти текст, який вимовляється та розпізнається. Чим суворіші ці правила, тим простіше реалізувати систему розпізнавання, і тим більш обмеженим буде набір речень, які вона здатна розпізнати.

4.2 Аудіовізуальне розпізнавання мови

У багатьох умовах функціонування (низька якість звукового сигналу, присутність сильного зовнішнього шуму або сторонніх розмов) стандартні системи автоматичного розпізнавання чи аналізу мови не можуть забезпечити необхідну якість роботи навіть при використанні різних методів фільтрації та шумозаглушення. Для того щоб підняти якість роботи цих систем застосовуються способи розпізнавання також візуальної інформації за допомогою технологій машинного зору, так створюючи системи цільного аудіовізуального розпізнавання мови з міксом «читання мови по губах». Очевидно, що мова передається не тільки у вигляді звукової хвилі, вона потрапляє від людини одночасно по декільком інформаційним каналам, по звуковому і візуальному. Наприклад часткові реалізації фонем дуже легко сплутати на слух (/ м / та / н /), та їх легко відрізнити візуально (/ м / промовляється з закритим ротом, а / н / - з відкритим). При сприйнятті мови людиною популярний також ефект МакГурка (McGurk) [13], коли правильний елемент з'являється тільки при поєднанні звукової та візуальної інформації.

Людина видає мову через одночасні дії декількох груп органів (трахея, голосові зв'язки, грудна клітка, легкі, гортанним трубка, порожнину глотки, порожнина носа, язик, піднебінна фіранка, порожнину рота, губи) [14]. Як бачимо, візуальні сигнали також дуже важливі для повного розуміння мовлення, бо дивлячись в обличчя співрозмовнику, легше розуміти його слова, особливо якщо мова іноземна. Так як сигнали від візуальних та слухових каналів дублюють, доповнюючи один одного.

Зараз існують два підходи до об'єднання звукової інформації з візуальною (information fusion) при бімомодальному розпізнаванні мови[15]:

– «раннє» об'єднання (early fusion) - у даному підході незалежно обчислюється параметричне представлення звукового і візуального сигналів, а потім, з урахуванням досить високого ступеня синхронності цих модальностей, формується єдиний вектор ознак (супер вектор) для кожного сегмента сигналу. На етапі аналізу мови використовуються методи, що використовують приховані Марковскі Моделі (ПММ) чи штучні нейронні мережі, тоді вони утворюють загальні моделі для акустичних мовних одиниць (фонем) та візуальних мовних одиниць (візем - зображень форми губ при проголошенні різних фонем);

– «пізднє» об'єднання модальностей (late fusion) – спосіб пізньої інтеграції використовує незалежні один від одного ПММ для імовірнісного моделювання звукових і візуальних мовних сигналів. Об'єднання модальностей можливо як на рівні станів імовірнісних моделей, так і на рівні потоків фонем / візем або навіть гіпотез розпізнавання фраз.

4.3 Сторонні сервіси розпізнавання мови

Існує величезна кількість сторонніх систем для роботи розпізнаванням мови. Хотілося б зупинитися на найбільш успішних.

Для обробки онлайн великих аудіо файлів :

– Speechmatics - великий словниковий запас в хмарі, вміщує в собі американську та великобританську англійську, висока точність розпізнавання.

– Vocapia Speech to Text API – не надто зручна, але високоякісна технологія.

Обробка офлайн:

– Speech Engine_IFLYTEK CO.,LTD. не найвідоміша китайська компанія, але вона постійно випереджає усіх у технологічних змаганнях.

– UWP – система розпізнавання мови від Microsoft для Windows Platform.

Open Source рішення:

– CMU Sphinx - Speech Recognition Toolkit - офлайн розпізнавання мови, через низьку потребу в ресурсах може бути використано на мобільному телефоні. OpenEars - Pocketsphinx на прошивці iOS, є також API для Node.js, Ruby, Java, Android.

– Kaldi - набір інструментальних засобів розпізнавання мови. UFAL-DSG/cloud-asr - Kaldi оснований на хмарній платформі iOS Speech Recognition - Kaldi адаптований для автономного розпізнавання по прошивці від Keen Research.

Також є сервіси від таких провідних компаній світу як Facebook, Google, Microsoft, деякі з них більш детально описані нижче.

4.3.1 Microsoft Speech Recognition

Дозволяє перетворювати усне мовлення в текст. Це API дозволяє включати і розпізнавати в реальному часі аудіо з мікрофона або іншого джерела, а також аудіо з файлу. У всіх цих випадках доступна можливість потокової передачі в реальному часі, завдяки якій безпосередньо в процесі відправки звуку на сервер повертаються результати розпізнавання.

Microsoft Speech Recognition API існує декількох видів:

– API розпізнавання мови Bing - рівень "безкоштовний": 5 000 безкоштовних транзакцій в місяць;

– API Bing для перетворення мови в текст фрази тривалістю до 15 секунд: $ 4 за 1000 транзакцій;

– API Bing перетворення тексту в мову: $ 4 за 1000 транзакцій.

4.3.2 Amazon Alexa Automatic Speech Recognition

Automatic Speech Recognition (ASR) - це технологія, яка перетворює вимовлені слова в текст. Коротше кажучи, це перший крок у забезпеченні реагування на голосові технології, такі як Amazon Alexa, коли ми запитуємо: «Алекса, що це за зовнішністю?

За допомогою ASR, голосова технологія може виявляти звукові сигнали і розпізнавати їх як слова. ASR є наріжним каменем всього досвіду голосу, дозволяючи комп'ютерам нарешті зрозуміти нас через нашу найбільш природну форму спілкування: мова.

Компанії прагнуть отримати знання як з існуючих каталогів, так і з їх вхідних даних. Переписуючи ці збережені носії, компанії можуть:

– аналіз даних виклику клієнтів;

– автоматизація створення субтитрів;

– цільова реклама на основі вмісту;

– увімкніть розширені можливості пошуку в архівах аудіо- та відео вмісту.

Ви можете легко розпочати роботу з транскрипцією за допомогою інтерфейсу командного рядка AWS (CLI), AWS SDK або консолі Amazon Transcribe.

З Amazon Lex, ви платите тільки за те, що ви використовуєте. З вас стягується на основі кількості текстових або голосових запитів, оброблених ботом, у розмірі 0,004 за голосовий запит і $ 0,00075 за текстовий запит. Наприклад, вартість 1000 запитів мовлення склала б 4,00 долара, а 1000 текстових запитів коштувала б 0,75 долара.

4.3.3 Google Speech Recognition API

Google Cloud Speech API дозволяє розробникам конвертувати аудіо в текст, застосовуючи потужні моделі нейронної мережі в простий у використанні API. API розпізнає понад 80 мов і варіанти, щоб підтримати вашу глобальну базу користувачів. Є можливість записати текст користувачів диктуючи мікрофон з додатку, включити командний контроль за допомогою голосу, або транскрибувати аудіофайли та багато іншого. Розпізнати аудіо закачаного в запиті, а також інтегрувати аудіо на власному носії з Google Cloud Storage, використовуючи ту ж технологію, Google використовує для запуску своїх власних продуктів.

Можливості Google Voice API:

– конвертація звуку у текст за допомогою нейронної мережі;

– 80+ мов;

– можливість зберігати файл та обробляти на сервері або обробка у реальному часі

– прибирає шуми

– може розпізнати слово з контексту речення

– будь-які пристрої (REST or gRPC request including phones, PCs, tablets and IoT devices)

– ціна використання: 0-60 хвилин щомісяця - безкоштовно, 61-1000000 хвилин - $0.006 за кожні 15 секунд. Це приблизно 25 центів за 10 хвилин аудіо.

4.3.4 Алгоритм Google Speech Recognition API

В основі актуальної версії голосового пошуку Google лежить покращений алгоритм для навчання нейронних мереж, створений спеціально для аналізу і розпізнавання акустичних моделей. В основу нових, рекурентних нейронних мереж (англ.: recurrent neural networks - RNN), лягла нейромережева темпоральна

класифікація (англ .: Connectionist Temporal Classification - CTC) і дискримінантний аналіз для послідовностей, адаптований для навчання подібних структур. Дані RNN набагато точніші, особливо в умовах сторонніх шумів, а головне - вони працюють швидше, ніж всі попередні моделі розпізнавання мови.

Раніше для розпізнавання та аналізу мови використовувалися «Моделі суміші (багатовимірних) нормальних розподілів» (англ .: Gaussian Mixture Model - GMM). Спочатку голосовий пошук Google також працював з цією технологією, поки не розробили новий підхід до перекладу звукових хвиль в осмислений набір символів, з якими може оперувати «класичний», текстовий пошук.

Переведення голосового пошуку Google на технологію Глибоких Нейронних Мереж (англ.: Deep Neural Networks - DNN) здійснив справжній прорив в області аналізу мови. DNN краще підходять для розпізнавання окремих звуків чи фонем, які вимовляються користувачем, ніж GMM, завдяки чому точність розпізнавання мовлення значно зросла.

У класичній системі розпізнавання голосу записаний звук ділиться на короткі (10 мс) фрагменти, кожен з яких потім аналізується на частоти, які містяться в ньому. Отриманий в результаті вектор характеристик проганяється через акустичну модель (наприклад, DNN), яка видає набір імовірнісних розподілів серед всіх можливих фонем. Прихована Марковська модель (часто використовувана в алгоритмах розпізнавання образів) допомагає виявити послідовні структури в цьому наборі розподілів ймовірностей.

Після цього дані аналізу об'єднуються з іншими даними, які надходять з альтернативних джерел інформації. Одним з них є Модель вимови (англ.: Pronunciation Model), яка з'єднує послідовність звуків в певні слова передбачуваної мови. (Прим.: Під «передбачуваною» мовою розуміється та мова, яка була обрана як «основна» в налаштуваннях голосового пошуку). Інше джерело - Мовна Модель: вона обробляє отримані слова і аналізує фразу цілком, намагаючись оцінити, наскільки ймовірна така послідовність слів у мові пошуку.

Далі вся інформація потрапляє в систему розпізнавання, який погоджує всю інформацію, щоб визначити фразу, яку вимовляє користувач. Наприклад, якщо

користувач вимовляє слово «museum», то його фонетична запис буде виглядати наступним чином: / m j u z i @ m /.

Точно сказати, де закінчився звук / j / і почався / u / може бути складно, але насправді для алгоритму це не важливо: головне, що всі ці звуки були вимовлянні.

Поліпшена акустична модель заснована на рекурентності Нейронних Мережах (RNN). Їх перевага полягає в тому, що вони мають цикли зворотного зв'язку в своїй топології, що дозволяють їм моделювати тимчасові залежності: коли користувач вимовляє / u / в попередньому прикладі, його мовний апарат одночасно виходить з процесу вимови попередніх звуків / j / i / m /. Слово «museum» при вимові виходить моментально, на одному видиху, і RNN можуть це розпізнати (рисунок 4.1).

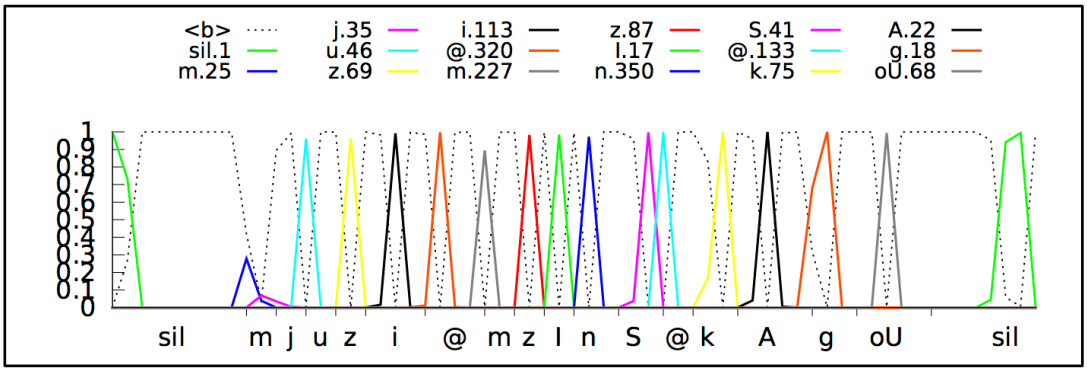

Рисунок 4.1 - Візуалізація роботи алгоритму Google Speech Recognition

RNN бувають різних типів, і для розпізнавання мовлення Google використовує спеціальні RNN з «довгою короткочасною пам'яттю» (англ. : Long Short-Term Memory - LSTM). Ці осередки пам'яті і складний механізм гейтов дозволяють LSTM RNN краще за інших нейронних мереж запам'ятовувати інформацію.

Використання навіть цих моделей вже значно поліпшило якість системи розпізнавання, наступним кроком стало навчання нейронних мереж розпізнаванню фонем у фразі без необхідності в постійному виділенні окремих «припущень» про імовірнісний розподіл кожної з них.

З нейромережевий темпоральной класифікацією (CTC) моделі навчилися виводити своєрідні «піки», які і відображають послідовність різних звуків в

звуковій хвилі. Вони можуть виділяти різні фонеми в правильній з точки зору мови послідовності звуків.

Найскладніше питання - це реалізувати розпізнавання фраз в реальному часі. Після безлічі спроб вдалося навчити потокові односпрямовані моделі обробляти більш протяжні звукові інтервали, ніж ті, що використовуються в «класичних» моделях аналізу та розпізнавання мови. У той час як самі обчислення відбуваються не так часто. При цьому витрати обчислювальних ресурсів насправді зменшилися, а швидкість роботи системи розпізнавання багаторазово збільшилася.

4.4 Досліди

У ході вибору технології для реалізації додатку були проведені певні досліди. Порівнювались Google, Microsoft Speech Recognition API та Amazon.

Першим дослідом було порівняння роботи кожного сервісу на різній відстані у кімнаті з мінімальним рівнем шуму (кімната для відеоконференцій), результати представлені у таблиці 4.1.

Таблиця 4.1 - Порівняння сервісів при мінімальному рівні шуму

Відстань до мікрофону	Google Speech Recognition API	Amazon ASR	Microsoft Speech Recognition API
~ 5 см	3 %	5 %	5 %
~ 1 м	5 %	5 %	6 %
~ 3 м	20 %	25 %	20 %

Наступним дослідом була перевірка сервісів у шумному місці (вулиця), також на різних відстанях для трьох обраних сервісів, результати зображено у таблиці 4.2.

У таблиці 4.3 можна побачити результати проведеного досліду з порівнянням розпізнавання окремого слова та цілого речення.

Таблиця 4.2 - Порівняння сервісів при високому рівні шуму

Відстань до мікрофону	Google Speech Recognition API	Amazon ASR	Microsoft Speech Recognition API
~ 5 см	5 %	8 %	5 %
~ 1 м	10 %	16 %	15 %

Таблиця 4.3 - Порівняння сервісів при різному об’ємі вхідних даних

Вхідні дані	Google Speech Recognition API	Amazon ASR	Microsoft Speech Recognition API
Речення	3%	3 %	5 %
Окреме слово	5 %	8 %	8 %

Можна побачити, що якість розпізнавання мови цілого речення вища, ніж окремого слова, це пов’язано з тим, що у цьому випадку слова розглядаються у контексті речення.

Також були зібрані відгуки від користувачів різних API на форумах, у персональних блогах, сайтах з навчальними матеріалами. За відгуками найбільш комфортна API вважається у Google. Тому, враховуючи попередні досліди та той факт, що додаток буде реалізований на Android, було вирішено зупинитися саме на цьому сервісі.

5 МОРФОЛОГІЧНИЙ АНАЛІЗ

Поточна стадія обробки складається з морфологічного та морфемного аналізу слів. Метою і результатом морфологічного аналізу є визначення морфологічних характеристик слова і його основної словоформи. Перелік всіх морфологічних характеристик слів і допустимих значень кожної з них залежать від природної мови. Проте, ряд характеристик присутні в багатьох мовах. Чому нам це потрібно? Бо фрази “запиши мене до лікаря” та “записати до лікаря” мають однаковий заклик до дії - це “записати”. Виділяючи морфологічні частини можно якісніше побудувати мапу токенів для лексичного аналізу.

5.1 Поняття морфології

Морфологія - розділ граматики, в якому вивчають явища, що характеризують граматичну природу слова як граматичної одиниці мови. Це вчення про будову та граматичні класи слів (частини мови), граматичні категорії і систему словозміни їх. Основною одиницею морфології є слово, але в аспекті граматичної будови, особливостей змінювання і творення, вираження властивих слову граматичних значень. Вхідним параметром є текстове представлення вихідного слова. Морфологія як наука передбачає розв'язання таких завдань: визначення принципів розчленування лексем на словоформи та об'єднання словоформ у лексеми; з'ясування частини семантики слова як морфологічної (граматичне значення); обґрунтування переліку морфологічних категорій та їх природи; опис сукупності формальних засобів, закріплених за відповідними частинами мови та їхніми морфологічними категоріями [16].

Граматична категорія - це найзагальніше поняття, що об'єднує ряд співвідносних граматичних значень і виражене в певній системі співвідносних граматичних форм. Поняття граматичної категорії ґрунтується на розумінні

об'єктивно існуючих взаємозв'язків між мовними системами і підсистемами. Граматична категорія є поняттям родовим щодо цілого ряду однорідних граматичних значень. Дієслівна категорія особи, наприклад, об'єднує ряд співвідносних граматичних значень, що виявляються у відповідних граматичних формах 1-ї, 2-ї, 3-ї особи; в категорії відмінка іменників узагальнюється вся різноманітність значень семи відмінків і система відмінкових форм.

Існують три основні підходи до проведення морфологічного аналізу:

– підхід «чіткої» морфології;

– підхід, який ґрунтується на деякій системі правил, по заданому слову визначають його морфологічні характеристики; на противагу до першого підходу його називають «нечіткої» морфологією;

– імовірнісний підхід, заснований на сполучуваності слів з конкретними морфологічними характеристиками.

Словник містить основні словоформи слів, для кожної з яких вказаний певний код. Відома система правил, за допомогою якої можна побудувати всі форми даного слова, відштовхуючись від початкової словоформи і відповідного їй коду. Крім побудови кожної словоформи, система правил автоматично ставить у відповідність їй морфологічні характеристики. При проведенні чіткого морфологічного аналізу необхідно мати словник усіх слів і всіх словоформ мови. Цей словник на вході приймає форму слова, а на виході видає його морфологічні характеристики.

При такому підході для проведення морфологічного аналізу заданого слова (рисунок 5.1) необхідно просто знайти його в словнику, де зберігаються точні, «остаточно відомі» значення всіх його морфологічних характеристик.

На жаль, цей метод можна застосовувати не завжди: слова, що надходять на вхід, можуть не входити в словник усіх словоформ. Така ситуація може виникнути через помилки введення початкового набору даних, або через наявність власних назв. Коли метод не дає потрібного результату, можно застосувати нечітку морфологія.

Рисунок 5.1 - Морфологічний аналіз на основі словника

У разі, коли не вдалося визначити характеристики слова за допомогою методів чіткої морфології, але вдалося розділити його на частини, то можна визначати морфологічні характеристики слова: можна побудувати систему правил, яка буде спиратися на наявність або відсутність будь-яких частин і видавати один або кілька припущень про морфологічних параметрах. Такий набір правил можна побудувати двома способами. Перший заснований на морфемному аналізі слів, що містяться в словнику всіх словоформ, і їх морфологічних характеристик. Розглянемо цю задачу формальні: відомі пари значень, що складаються з морфемної будови слова і його морфологічних характеристик. Це є не що інше, як «вхід» і «вихід» системи правил, яка за морфемного будовою слова визначатиме його морфологічні характеристики. Завдання побудови такої системи правил можна вирішити за допомогою самонавчання системи (рисунок 5.2).

Рисунок 5.2 - Нечіткий мофрологічний аналіз

Інший підхід полягає у формуванні набору правил власноруч. Його реалізація - написання експертної системи діагностуючого типу.

Імовірнісний спосіб проведення морфологічного аналізу слів полягає в наступному. Одна і та ж словоформа може належати відразу до декількох граматичними класами. Для кожної словоформи визначаються всі її граматичні класи, а також ймовірність її ставлення до кожного з цих класів. Це виконується на основі деякого набору документів, де кожному слову попередньо поставлений у відповідність граматичний клас. Після цього обчислюються ймовірності поєднань певних граматичних класів для слів, що стоять поруч. На основі цих чисел може проводитися аналіз слів, але для нього необхідно вже не тільки саме слово, але і слова, які стоять поруч з ним.

В ході автоматичного морфологічного аналізу документу природною мовою обробляється кожне слово тексту, то вирішуються такі завдання:

– стемінг;

– лематизація;

– встановлення морфологічних ознак слова (граммем).

Розглянемо ці завдання більш детально, але перед цим визначимо деякі поняття. Під лексемою (lexeme) будемо розуміти одиницю словникового складу мови в сукупності всіх його конкретних граматичних форм. Лексема - абстрактне поняття, яке об'єднує всі можливі форми одного слова (словоформи, word forms). Лема (Lemma) - канонічна (нормальна, словникова) форма лексеми. Лема відповідає формі слова, яка наводиться в будь-якому словнику. В українській та російській мовах словниковій формі іменника відповідає форма в називному відмінку в однині (брат), для прикметників словникова форма буде відповідати формі називного відмінка, чоловічого роду, однини (розумний), у дієслів словникова форма відповідає інфінітива (бігти). Лематизація (lemmatization) - це процес встановлення лем для слів документу.

Стемінг (stemming) - процес пошуку псевдооснови (stem) слова, деякою незмінної частини всіх словоформ, яка, взагалі кажучи, не завжди може збігатися з його коренем. Завдання стемінг актуальна при аналізі мов з досить розвиненим словозміненням. Лематизація і стемінг мають одну мету – звести різноманітність форм одного слова до одного інваріанта, що необхідно, наприклад, при вирішенні

задач інформаційного пошуку або більш загальної задачі порівняння текстів, коли потрібно ототожнювати всі форми слова.

Встановлення морфологічних ознак слів та фраз тексту необхідно для зняття багатозначності результатів лематизації і для подальшого - синтаксичного аналізу. Задача встановлення морфологічних ознак може розглядатися як завдання розмітки або тегування (tagging) документу - встановлення тегів (морфологічних ознак) словами текстів. Набір встановлюваних ознак залежить від мови. Так, в українській та російській мовах встановлюється частина мови, для іменників і прикметників - число, рід і відмінок, для дієслів - форма, час і спосіб тощо, а в англійській мові, де відсутні граматичні відмінок і рід, часто обмежуються встановленням частини мови. У дослідженнях цей процес називають Part-Of-Speech Tagging (POS Tagging). Встановлення морфологічних ознак зазвичай виконується спільно з лематизацією.

При вирішенні завдань морфологічного аналізу дуже часто проявляється неоднозначність. Так, слова різних частин мови можуть мати однакове написання (мила - іменник і дієслово), однієї словоформи може відповідати кілька лем (наприклад, словоформи мила відповідають леми мити і мило), одна словоформа може виражати кілька морфологічних ознак (приклад відмінникової багатозначності - збіг форм називного і знахідного відмінка для деякого класу слів української мови: стіл, екран), але в результаті морфологічного аналізу у слова повинні залишитися тільки однозначні грамеми. Зняття морфологічної багатозначності виконується зазвичай в окремій від морфологічного аналізу процедурі, цей процес описаний в наступному розділі

5.2 Стемінг

Методи стемінгу засновані на відсікання афіксів. Найбільш відомим методом стемінгу є Алгоритм Портера [16], який заснований на послідовному застосуванні правил перетворення закінчень слів. Правила мають такий вигляд:

(умова): S1 → S2, що означає, що якщо слово закінчується постфіксом S1, і ліва частина перед S1 задовольняє заданій умові, то S1 необхідно замінити

Умова в правилі містить, наприклад, перевірку на кількість повторень послідовності голосна-приголосна, наявність голосної букви, закінчень подвійну приголосну тощо. Алгоритм складається з семи послідовно виконуваних кроків, на кожному з яких застосовується певне безліч правил. На рисунку 5.3 наведено приклади правил першого кроку[16].

SSES → SS	*caresses* → *caress*
IES → Y	*ponies* → *pony*
SS → SS	*caress* → *caress*
S → ε	*cats* → *cat*

Рисунок 5.3 – Правила першого кроку

Правила в алгоритмі Портера трансформуються в кінцеві автомати, що робить його дуже швидким. Переваги всіх стемерів - висока швидкість, недолік - низька точність. Стемінг погано підходить для пошуку інваріантної форми слова, тому що для різних слів породжуються одні псевдо основи (для люб-ити, люб-ов буде отримана одна основа люб).

5.3 Зняття морфологічної багатозначності

Найбільшу проблему для аналізу текстів на природних мовах представляють омоформи - форми слів, що мають однакове написання. Форми можуть збігатися у словах різних частин мови, наприклад, словоформа мила є формою дієслова мити і іменника мило. Буває і збіг форм однієї лексеми в деяких граматичних значеннях, наприклад, мами відповідає формі називного відмінка множини і формі родового відмінка однини слова мама. Завдання зняття морфологічної неоднозначності полягає в загальному випадку у виборі правильного варіанту омоніми, тобто однозначного встановлення частини мови,

леми і морфологічних ознак у Омоформи. Для англійської мови завдання зводиться до вибору частини мови і називається part-of-speech disambiguation.

Підходи до вирішення морфологічної неоднозначності засновані на аналізі контексту (оточуючих слів) неоднозначного слова і поділяються на статистичні (statistical) і засновані на правилах (rulebased) [17]. Правила можуть складатися вручну або виводитися по розміченим корпусам, статистичні методи засновані на навчанні за великими розміченими корпусами. Методи дозволу морфологічної неоднозначності застосовуються зазвичай після первинної розмітки, що виконується за допомогою словників або іншими методами.

5.4 Порівняння систем морфологічного аналізу

Все морфо процессоры надають найбільш важливу для української та російської мов функцію лематизації словоформ, при цьому зі зняттям омонімії. Ця функція реалізується і для позасловникових слів. Функція стемінгу є менш популярною в реалізаціях тому, що менш потребувана на практиці, проте все процесори, крім TreeTagger, надають можливість отримання словозмінної парадигми заданої словоформи, а з її допомогою досить просто отримати псевдооснову слова. Морфологічний синтез також реалізований лише в двох з розглянутих процесорів, хоча в багатьох задачах комп'ютерної лінгвістики дана функція є важливою. Порівняльна характеристика чотирьох популярних процесорів зображена у таблиці 5.1.

Два з представлених процесорів є закритими і поширюються виключно у вигляді бінарних файлів. Словник MyStem є закритим, словник TreeTagger доступний у вигляді бінарного файлу. Швидкість оброблюваних слів у всіх процесорів є досить високою. Як правило, істотне уповільнення обробки спостерігається на більш пізніх етапах аналізу природних мов, тому швидкість морфопроцессора рідко стає вузьким місцем. Можливість підключення словника є

особливо важливою для задач обмежених предметних областей. Дану функцію надає MyStem.

Таблиця 5.1 – Порівняння систем морфологічного аналізу

Система	AOT	MyStem	TreeTagger	Pymorphy
Відкритий код	так	ні	ні	так
Швидкість слів у секунду	60-90 тис.	100-120 тис.	20-25 тис.	80-100 тис.
Підключення сторонніх словників	ні	так	так	ні
Об’єм словника	160 тис.	250 тис.	210 тис.	250 тис.

Великою проблемою, пов'язаною з морфологічними процесорами, є використання власної системи морфологічних тегів в кожному з них. Через невідповідність морфологічних тегів складно порівнювати роботу процесорів, оцінювати їх точність і повноту на розмічених корпусах. Рішенням даної проблеми міг би бути універсальний конвертер з однієї системи тегів в іншу, який відсутній в усіх розглянутих аналізаторах.

6 СИНТАКСИЧНИЙ АНАЛІЗ

Після того як зроблений аналіз кожного слова, починається аналіз окремих речень (синтаксичний аналіз), що дозволяє визначити взаємозв'язок між окремими словами і частинами речення.

6.1 Поняття синтаксису

Синтаксис — розділ граматики, що вивчає граматичну будову словосполучень та речень у мові.

Синтаксис можна розмежовувати на:

– синтаксис словосполучень, який встановлює синтаксичні властивості окремих слів як частин мови, тобто правила їхньої сполучуваності з іншими словами;

– синтаксис речень, спрямований на дослідження типів, ознак речень, зв'язків слів і сполук у складі речень і висловлювань.

Словосполучення — найпростіша синтаксична одиниця мови, утворена з двох або кількох повнозначних слів, пов'язаних між собою в граматичному плані і за змістом. Наприклад: червоний ліхтарик, стояти вгорі, дивитися на екран.

Речення — це синтаксична одиниця, що виражає певну думку, має інтонаційну завершеність і служить для спілкування (він довго збирався це зробити, але не встиг).

Завданням синтаксичного аналізу є явний опис синтаксичної структури речення. Такий опис може бути виконано за допомогою структур залежностей, або за допомогою структур складових. Перший підхід передбачає, що слова в реченні пов'язані синтаксичними відносинами або зв'язками. Інтуїтивно можна вважати, що пов'язані один з одним такі пари слів, з яких одне в якомусь сенсі «визначає» або «доповнює» інше [18]. Наприклад, слово маленькі, що відноситься до слова діти,

«визначає» або «уточнює» його значення тощо. Другий підхід заснований на припущенні, що слова в реченні об'єднуються в групи, всередині яких слова тісно пов'язані.

Першочерговою задачею синтаксичного аналізу є задача сегментації речення. Сегментом є частина речення, що виділена знаками пунктуації та описує окрему ситуацію. В сегменті виділяється його предикативна вершина, виражена в більшості випадків фінітною формою дієслова або іншим предикативним словом (прислівник, дієприслівник). Задача синтаксичного аналізу вирішується на основі різних методів формальних граматик, які встановлюють певні правила композиції синтаксичних структур. Найчастіше використовують методи машинного навчання.

6.2 Структури залежностей

У синтаксичному зв'язку завжди одне слово є головним, а інше - залежним. Таким чином синтаксичний зв'язок завжди спрямований, а між двома словами не може бути двох (різноспрямованих) зв'язків. Синтаксичні відносини є асиметричними і не транзитивними.

Структуру залежностей речень можна уявити у вигляді дерева залежностей (dependency tree), в вузлах якого розташовані слова (словоформи), а орієнтовані дуги між вузлами у вигляді стрілок, які відображаються зазвичай над реченням, позначають синтаксичні залежності. Стрілки можуть позначатися типами зв'язків. У кореневому вузлі синтаксичного дерева знаходиться дієслово, який займає центральне положення в структурі речень. Дерево залежностей для речення «Голубая чашка стоит на столе» представлено на рисунку 6.1.

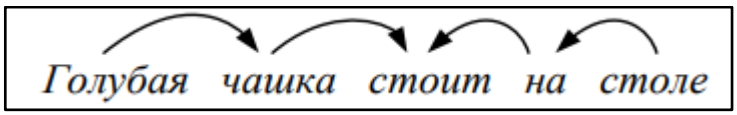

Рисунок 6.1 - Дерево залежностей для речення

Вважається, що формалізм залежностей добре відображає специфіку мов з довільним порядком слів, в яких між словами може бути присутнім значна кількість непроективних зв'язків. До таких мов відноситься німецька, чеська, українська, російська, а також інші східнослов'янські мови. В українськійй та російській комп'ютерній лінгвістиці структура залежностей є основним засобом представлення синтаксису при автоматичному синтаксичному аналізі текстів.

6.3 Формальний опис синтаксису

Для автоматичного синтаксичного аналізу природних мов використовуються формальні граматики, які містять формалізовані правила побудови правильних висловлювань на мові, при цьому висловлювання є ланцюжки символів (слів). Формальні граматики складаються з правил групування та впорядкування символів мови, і лексикону слів і символів.

Символи, з яких складаються ланцюжки, називаються термінальними символами (terminal symbols), вони утворюють основний алфавіт. Символи для визначення класу ланцюжків символів називаються нетермінальними символами (non-terminal symbols), які утворюють допоміжний алфавіт. Правила мають вигляд x → y, що означає «замінити x на y», де x і y - будь-які ланцюжки символів.

Побудуємо найпростішу граматику для української мови. Правила

NP → Noun // чашка

NP → Adjective Noun // блакитна чашка

задають іменні групи, що складаються з іменників і поєднань «прикметник + іменник».

Правило

PP → Preposition NP // на столі задає прийменникові групи, а правила

VP → Verb NP // читати книгу

VP → Verb NP PP // покласти книгу на стіл

VP → Verb PP // стоїть на столі

задають дієслівні групи. Базовий алфавіт задається наступним чином:

Noun → чашка | книга | стіл ...

Adjective → блакитний | Вродливий …

Preposition → на | в ...

Verb → стояти | лежати | покласти

Особливий нетермінальний символ, званий початковим символом, позначається S і відповідає всьому ланцюжку. правило

S → NP VP

це задає безліч речень, що складається з простих синтаксичних груп.

6.4 Методи синтаксичного аналізу

Синтаксичний аналіз поділяється на поверхневий (shallow) і повний (deep). Поверхневий аналіз (chunking) призначений для виділення смислових складових, таких як іменникова група, дієслівна група, прикметникова група. Повний синтаксичний аналіз являє собою структуру речення у виді синтаксичного дерева. Сьогодні розроблені різноманітні формальні граматичні теорії синтаксису: граматика залежностей, граматика безпосередніх складових, категоріальна граматика, лексико-функціональна граматика.

Поверхневий синтаксичний аналіз охоплює такі завдання як поділ речення на рекурсивно невкладені синтаксичні групи, сегментацію (виділення в реченні різних оборотів і простих речень у складі складного), а також побудова поверхневого синтаксичного дерева. Поверхневі аналізатори зазвичай не призначені для встановлення всіх синтаксичних зв'язків у реченні, не враховують далекі зв'язки і не призначені для визначення граматичних функцій слів речення. Подібне спрощення завдання синтаксичного аналізу в порівнянні з глибоким аналізом дозволяє використовувати обчислювально і алгоритмічно більш прості

методи. Крім того, в рамках спрощеної завдання вдається досягти високих показників якості.

6.5 Системи синтаксичного аналізу

Для паралельного виконання процесів з попередньої глави ми можемо використовувати сучасний підхід з використанням нейронних мереж. Так проаналізувавши стан речей було вибрано SyntaxNet. SyntaxNet — це нейронна мережа з відкритим кодом, реалізована в TensorFlow, що є основою для систем нейронних мереж (NLU). Один з найточніший у світі аналізаторів з відкритим кодом, розроблений Google. Сам фреймворк включає в себе весь код, необхідний для підготовки нових моделей SyntaxNet на власних даних, а також Parsey McParseface - це англійський парсер, який можна використовувати для аналізу англійського тексту.

Parsey McParseface побудований на потужних алгоритмах машинного навчання, які навчаються аналізувати лінгвістичну структуру мови, і які можуть пояснити функціональну роль кожного слова в даному реченні. Оскільки Parsey McParseface є найточнішою такою моделлю у світі, то використовується у автоматичному вилученні інформації, перекладах та інших основних додатках NLU.

Він працює з безліччю язиків, але немає української моделі. Тому будемо використовувати надбудову над SyntaxNet у зв’язці з DRAGNN.

TensorFlow DRAGNN (Динамічна повторювана ациклічна графічна нейронна мережа) - це набір інструментів для побудови та вивчення повністю динамічних графіків нейронних обчислень у TensorFlow. На відміну від традиційної рекурсивної нейронної мережі (наприклад, моделі динамічного дозування, такі як TensorFlow Fold), DRAGNN використовує вивчену політику для додавання повторюваних ребер до графіка обчислення на льоту як функції входу і виходу мережі.

Чому нам потрібні повністю динамічні графіки обчислення? Як виявилося, такі моделі виникають природно при використанні глибокого навчання для побудови багаторазових компонентів. Наприклад, розуміння природної мови можна розділити на окремо корисні стадії:

– модуль сегментації спочатку розбиває рядок символів на дискретні слова разом з векторними зображеннями цих слів;

– модуль позначки позначає кожне слово властивостями, такими як частина мови, морфологічні мітки або посилання на зовнішні бази знань, оскільки вказує обґрунтоване знання у векторне представлення;

– модуль розбору використовує словосполучення для побудови фраз і генерує векторні представлення для кожної фрази;

– модулі міркувань на рівні речення говорять про значення фраз.

Наприклад, настрої або семантичні ролі, і конструюють семантичні представлення речення.

Після запуску ми отримаємо наступні результати, ми порівнюємо оригінальний SyntaxNet (original) та розширений з DRAGNN (upgraded) (таблиця 6.6.1)

Як бачимо з результатів порівняння SyntaxNet показує гірші резултати ніж SyntaxNet + DRAGNN. SyntaxNet є основою для того, що відомо в академічних колах як синтаксичний аналізатор, який є ключовим першим компонентом у багатьох системах NLU.

Бібліотека DRAGNN забезпечує рамки для визначення та навчання такого конвеєра в одному графіку TensorFlow. Цей графік обов'язково повинен бути динамічним для того, щоб враховувати дискретні проміжні структури, такі як дерева розбору, фрази та проміжки. Крім того, бібліотека дозволяє навчати спільні моделі в багатозадачних рамках, навіть якщо анотації для окремих проміжних завдань відсутні.

Враховуючи речення в якості вхідних даних, він позначає кожне слово тегом «мова» (POS), який описує синтаксичну функцію слова, і визначає синтаксичні відносини між словами у реченні, представленими в дереві розбору залежностей.

Ці синтаксичні відносини безпосередньо пов'язані з основним значенням даного речення.

Таблиця 6.1 - Порівняння SyntaxNet та розширений з DRAGNN

	Language	**Ukrainian**	**English**	**English-LinES**	**Russian**	**Spanish**
No. tokens	original	-	25096	8481	9573	7953
POS, %	original	-	90.48	95.34	95.27	95.27
fPOS, %	original	-	89.71	93.11	95.02	-
Morph,%	original	-	91.30	-	87.75	95.74
UAS, %	original	-	84.79	81.50	81.75	85.06
	upgraded	72.19	87.60	82.43	85.18	90.32
LAS, %	original	-	80.38	77.37	77.71	81.53
	upgraded	62.79	84.20	78.46	80.71	87.16

Однією з головних проблем, яка робить розбір складним, є те, що людські мови демонструють помітні рівні неоднозначності. Це не важко для речень помірної довжини - скажімо, 20 або 30 слів - але коли сотні, тисячі або навіть десятки тисяч можливих синтаксичних структур. Синтаксичний аналізатор природних мов повинен якось шукати всі ці альтернативи і знаходити найбільш вірогідну структуру з урахуванням контексту. Як дуже простий приклад: “Аліса проїхала по вулиці в її машині” - і це речення має принаймні два можливих розбори залежностей:

Перший відповідає (правильному) тлумаченню, коли Аліса їде в машині; друга відповідає (абсурдному, але можливому) тлумаченню, де вулиця знаходиться в її машині. Неоднозначність виникає тому, що прийменник може змінювати або водіння, або вулицю; цей приклад є прикладом того, що називається приймальною фразою невизначеності прикріплення.

Люди легко працюють з двозначністю, майже до того моменту, коли проблема непомітна; проблема полягає в тому, щоб комп'ютери робили те саме. Такі багатозначні невизначеності, як у більш тривалих висловлюваннях, утворюють комбінаторний вибух у кількості можливих структур для пропозиції. Зазвичай переважна більшість цих конструкцій є дико неправдоподібними, але, тим не менш, є можливими і повинні якось відкидатися синтаксичним аналізатором.

SyntaxNet застосовує нейронні мережі з проблемами неоднозначності. Вхідне речення обробляється зліва направо, при цьому враховуються залежності між словами, які поступово додаються як кожне слово у реченні. Нейронна мережа дає оцінки для конкуруючих рішень на основі їх достовірності. З цієї причини дуже важливо використовувати в моделі модель пошуку пучка. Замість того, щоб просто прийняти перше рішення в кожній точці, множинні часткові гіпотези зберігаються на кожному кроці, причому гіпотези лише відкидаються, коли існує кілька інших більш вірогідних гіпотез.

7 МЕТОДИ ОЦІНЮВАННЯ РОБОТИ СИСТЕМ

7.1 Кількісна оцінка систем розпізнавання мови

Існують різні за складністю і прикладному значенню завдання розпізнавання:

– ізольованих слів (команд);

– ключових слів в потоці мовлення;

– зв'язного мовлення (точне обговорення фрази з паузами між словами);

– злитої промови (розділяють диктовку у вузькій тематичної області, і спонтанну мову, наприклад, в діалозі між людьми).

Оцінка системи, яка розпізнає окремі команди, не представляє будь-яких труднощів - кількість неправильно розпізнаних команд ділиться на загальну кількість випробувань і виходить відсоток помилки. Для систем, які розпізнають злиту мову, ситуація складніша.

Однією з основних проблем в роботі систем автоматичного розпізнавання мови є об'єктивне кількісне оцінювання результатів розпізнавання, що має важливе значення як для інженерів, так і для кінцевих користувачів систем. Методологія кількісного оцінювання продуктивності призначена для порівняння і зіставлення різних систем розпізнавання, в ній виділяють критерій, показник і метод:

– критерій - це область оцінювання, тобто то, що необхідно оцінити: наприклад, точність розпізнавання мови, швидкість її обробки, роботоспосібність тощо;

– показник (міра або метрика) визначає конкретну властивість, яка оцінюється для обраного критерію: наприклад, відсоток правильно розпізнаних слів, час обробки сигналу, рівень максимально допустимого шуму при збереженні працездатності тощо.

– метод - це спосіб визначення відповідного значення для даного показника: наприклад, порівняння розрізнених слів з послідовністю сказаних слів, оцінка часу обробки.

Зазвичай при розробці систем автоматичного розпізнавання мови використовуються три різних набори даних: навчальний ("train"), оцінний ("dev") і оціночний/тестовий ("eval").

Навчальний набір даних (зазвичай це найбільша частина мовних даних) використовується тільки для створення і навчання моделей системи. Налагоджувальний набір даних використовується для налаштування та адаптації параметрів автоматичної системи перед фінальною стадією оцінки, цей набір даних повинен мати такий формат, що і тестові дані. Оціночні дані містять мовні дані, які не використовувалися для навчання та налаштування системи, і доступні тільки при фінальній оцінці системи. Виділяють два основних критерії при оцінці роботи систем розпізнавання мови, які далі розглянути детально: якість розпізнавання і швидкість обробки [19].

7.2 Показники точності розпізнавання мови

Точність розпізнавання – основний показник якості у системах автоматичного розпізнавання мови, який визначається як відсоток правильно розпізнаних слів (WRR - Word Recognition Rate) або, навпаки, неправильно розпізнаних слів (WER - Word Error Rate). Іноді також використовується показник помилок розпізнавання фраз / речень (SER - Sentence Error Rate), який є важливим в діалогових системах, де коригування гіпотези розпізнавання неможливе навідміну від завдання диктування. Останнім часом в якості основного показника точності роботи систем розпізнавання мови використовується показник WER, а саме, його абсолютне значення або відносне, якщо порівнюються різні моделі / системи.

Оскільки з розвитком мовних технологій показник WER все більш наближається до нуля, то поліпшення його значення більш наочно, ніж підвищення точності розпізнавання слів. Метод визначення показника WER складається у вирівнюванні двох текстових рядків (перший - це результат розпізнавання, а другий

- запис того, що було сказано в дійсності) за допомогою алгоритму динамічного програмування з обчисленням відстані Левенштейна [20]. Відстань Левенштейна - "вартість" редагування даних (мінімальна кількість або зважена сума операцій редагування [21]) для перетворення першого рядка в другий (7.1):

$$\mathrm{WER} = \frac{S + D + I}{T}, \quad \mathrm{WRR} = 1 - \mathrm{WER}, \tag{7.1}$$

де S - число операцій ручної заміни,

D – число операцій видалення,

I - число операцій вставки;

T - кількість слів у розпізнається фразі.

Для оцінювання результатів автоматичного розпізнавання мови також використовується такий показник, як відсоток коректно розпізнаних слів (WCR - Word Correctly Recognized), який не враховує помилкові вставки слів, зроблені системою.

WER - інтуїтивно зрозумілий показник якості розпізнавання для аналітичних мов з досить простою морфологією, в яких граматичні значення однозначно виражаються самим словом (наприклад, англійську або французьку). Однак синтетичні мови (наприклад, аглютинативні фінська, турецька або флективні українська, російська) мають багату морфологію словотворення; в деяких азіатських мовах (китайська, корейська) використовуються склади замість слів; в тайській мові відсутні явні роздільники кордонів слів. Тому ці мови можуть синтезувати досить довгі осмислені словоформи з декількох частин (морфем), що визначають граматичні ознаки. Зазвичай кінець словоформи вимовляється в звичайному мовленні не так чітко, як початкова частина слова, що призводить до акустичної невизначеності і в середньому до більш високих порівняно з аналітичними мовами значенням показника WER.

У синтетичних мовах для оцінювання точності автоматичного розпізнавання мови можуть застосовуватися інші показники: помилки розпізнавання букв / символів, фонем (звуків мови), складів або морфем [21]. Крім

того, для деяких синтетичних мов (української та російської) адекватним їх структурі показником є флективна помилка розпізнавання слів (IWER - Inflectional Word Error Rate) [22], яка визначається наступним чином (7.2):

$$\text{IWER} = \frac{S_{\text{hard}} \cdot C_{\text{hard}} + S_{\text{soft}} \cdot C_{\text{soft}} + D + I}{T}, \quad C_{\text{soft}} < C_{\text{hard}}, C_{\text{hard}} \geq 1, 0 \leq C_{\text{soft}} < 1. \tag{7.2}$$

де Shard - кількість помилок,

Csoft – кількість негрубих помилок;

Показник IWER приписує вагу Chard всім невірним замінам слів, які призводять до заміни лексеми слова, тобто до грубих помилок розпізнавання і меншу вагу Csoft - всім негрубим помилкам в словах, де було невірно розпізнано закінчення словоформи, але основа слова розпізнана правильно.

При оцінюванні точності автоматичного розпізнавання мови за показником WER передбачається, що всі слова у вхідній фразі однаково інформативні та важливі. Однак очевидно, що в системах, відмінних від диктування, наприклад в діалогових або в системах розуміння (сенсу) мови, деякі значущі (ключові) слова важливіші, ніж інші (функціональні слова, прийменники). В роботі запропоновано оцінювати точність розпізнавання, використовуючи зважений показник неправильно розпізнаних слів (WWER - Weighted Word Error Rate), який визначається за формулою (7.3):

$$\text{WWER} = \frac{V_S + V_D + V_I}{V_T}, \tag{7.3}$$

$$V_T = \sum_{W_i \in T} v_{W_i}, V_I = \sum_{\hat{W}_i \in I} v_{\hat{W}_i}, V_D = \sum_{W_i \in D} v_{W_i}, V_S = \sum_{s_j \in S} v_{s_j}, v_{s_j} = \max\left(\sum_{\hat{W}_i \in s_j} v_{\hat{W}_i}, \sum_{W_i \in s_j} v_{W_i} \right), \tag{7.4}$$

де *vWi* - вага слова *Wi*, яке є *i*-м у вхідній фразі,

vW^i - вага слова *W^*, яке є і-м в гіпотезі розпізнавання,

s j - *j*-й замінений фрагмент фрази (або одне слово),

v_{sj} - вага даного фрагмента *s j*.

Таким чином, згідно з показником WWER кожне слово може мати різну вагу відповідно до його впливу на подальше розуміння сенсу сказаної фрази.

Національним інститутом стандартів і технологій (NIST, США) нещодавно був запропонований такий показник, як кількість неправильно розпізнаних слів у мові кожного з дикторів (SAWER - Speaker Attributed Word Error Rate) - для завдання стенографування нарад в яких передбачається участь декількох дикторів.

Дане завдання об'єднує технології автоматичного розпізнавання мови і діарізації голосу диктора (розмітки звукового сигналу на фрагменти "хто і коли говорив" - "Who Spoke When") [23]. Результатом цієї об'єднаної системи є текстова транскрипція вхідного одноканального звукового сигналу для кожного розпізнаного слова з явним зазначенням на мовця.

Однак слід розуміти, що відсоток неправильного розпізнавання – це тільки кількісний показник точності розпізнавання, але не ймовірність розпізнавання слова у фразі, так як показник WER не обмежується інтервалом ймовірності [0; 1] і не має верхньої межі.

Наприклад, уявімо, що хтось вимовив фразу, що складається з 10 слів, але система її повністю розпізнала неправильно і запропонувала гіпотезу з 15 інших слів. В цьому випадку WER = 150% (S = 10, I = 5, H = D = 0), і, отже, показник точності WRR негативний (Тобто -50%), що не має сенсу. Для того щоб вирішити цю проблему, нещодавно були запропоновані інші показники, зокрема: помилка розпізнавання відповідностей (MER – Match Error Rate) і показник втрати інформації, що міститься в словах (WIL - Word Information Lost) [24], засновані на величині відносної втрати інформації і визначаються наступним чином (8.5):

$$\mathrm{MER} = \frac{S + D + I}{T_P = H + S + D + I} = 1 - \frac{H}{T_P}; \quad \mathrm{WIL} = 1 - \frac{H^2}{T \cdot T_O}, \text{ если } H >> S + D + I, \tag{7.5}$$

де To - кількість слів у гіпотезі розпізнавання.

Проте обидва цих показника рідко застосовуються, так як забезпечують зазвичай дещо меншу точність розпізнавання по порівняно зі стандартними показниками.

Всі перераховані вище показники враховують тільки одну найкращу гіпотезу розпізнавання кожної фрази. Однак деякі системи автоматичного розпізнавання мови (наприклад, фонетичний декодер) здатні видавати відразу кілька гіпотез розпізнавання з найбільшими можливостями - так званий список N кращих гіпотез (N-best List).

Додатковим показником для оцінки таких результатів є показник помилок розпізнавання слів в списку кращих гіпотез, який оцінюється шляхом вибору з N гіпотез, ранжируваних по зменшенню оцінки правдоподібності, єдиною гіпотези, що має найменший рівень помилок. Показник WER гіпотези з мінімальним рівнем помилок по кожній вхідний фразі вибирається як основний результат розпізнавання, що характеризує відсоток помилок розпізнавання в списку N кращих гіпотез [25].

7.3 Показники швидкості розпізнавання мови

Другий важливий критерій роботи системи автоматичного розпізнавання мови - швидкість обробки мови. Швидкість обробки обчислюється, як правило, з використанням міри, яка називається показником швидкості (SF - Speed Factor) і також відомої як показник реального часу (RT - Real Time), який визначається відношенням загального часу обробки, необхідного для аналізу всієї записаної мови на одному ядрі процесора, до тривалості вихідного аналізованого аудіосигналу.

Наприклад, якщо 10-хвилинний аудіофайл обробляється системою розпізнавання мови протягом 5 хвилин, то SF = 0,5 RT, якщо файл обробляється протягом 20 хвилин, то SF = 2,0 RT, що значно гірше. Швидкість обробки може бути також вказана в абсолютних значеннях часу (наприклад, кількість хвилин / секунд для обробки вхідного сигналу), проте це не є наочним. Іншим показником швидкості автоматичного розпізнавання мови може бути період очікування обробки відліку (SPL - Sample Processing Latency). Цей показник означає

максимальну кількість аудіо, яку алгоритм розпізнавання повинен обробити до видачі результату про перший відлік сигналу.

При створенні системи автоматичного розпізнавання мови, яка володіє великим словником і працює в реальному часі з використанням мікрофона, часто потрібно знайти компроміс між точністю розпізнавання і швидкістю обробки.

Налаштування деяких параметрів системи може поліпшити точність розпізнавання, але зменшити швидкість обробки. У цьому випадку може бути корисним графік залежності показника WER від швидкості розпізнавання в деяких контрольних точках; результати аналізу цього графіка дозволяють вибрати оптимальні параметри системи.

8 МЕТОД ОПОРНИХ ВЕКТОРІВ У ЗАДАЧАХ КЛАСИФІКАЦІЇ

В якості алгоритму навчання можна зупинитися на методі опорних векторів (SVM).

В машинному навчанні метод опорних векторів — це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами (ОВМ, англ. support vector machines, SVM, також опорно-векторними мережами, англ.support vector networks). Для заданого набору тренувальних зразків, кожен із яких відмічено як належний до однієї чи іншої з двох категорій, алгоритм тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої категорії, роблячи це не ймовірнісним бінарним лінійним класифікатором. Модель ОВМ є представленням зразків як точок у просторі, відображених таким чином, що зразки з окремих категорій розділено чистою прогалиною, яка є найширшою. Нові зразки тоді відображуються до цього ж простору, й робиться передбачення про їхню належність до категорії на основі того, на який бік прогалини вони потрапляють.

Використання методу опорних векторів і формалізація завдання класифікації визначаються типом кривої (поверхні) в просторі, тобто класифікатором. Найбільш простий і наочний - це випадок лінійного класифікатора, коли роздільна поверхня являє собою площину. Більш трудомістким є випадок нелінійного класифікатора.

Лінійний класифікатор – це перший спосіб вирішення задачі класифікації. Розглянемо найпростішу задачу бінарної класифікації: є колекція сірих і чорних точок на площині. Необхідно знайти таке правило, за яким нову, незабарвлену точку можна було б пофарбувати в один з цих двох кольорів. Ідея полягає в наступному: знайти пряму, яка відокремлює всі сірі точки від чорних точок. Якщо вдасться знайти таку пряму, то класифікувати кожну нову точку можна буде наступним чином: якщо точка лежить вище прямої, то вона сіра, якщо нижче -

чорна. Формалізуємо цю класифікацію: необхідно знайти вектор w такий, що для деякого граничного значення b і нової точки x виконується умова (8.1 та 8.2):

$$w \cdot x_i > b \Rightarrow y_i = 1 \tag{8.1}$$

$$w \cdot x_i \bullet b \Rightarrow y_i = -1 \tag{8.2}$$

Якщо скалярний добуток вектора w на x_i більше допускає значення b, то нова точка належить першій категорії, якщо менше - другій. Насправді вектор w перпендикулярний до роздільної прямої, а значення b залежить від найкоротшої відстані між роздільною прямою і початком координат.

Розглянемо приклад на рисунку 9.1. Для прямої L_2 межа b дорівнює 0, а для прямої L_1 – довжині перпендикуляра, опущеного на L_1 з початку координат.

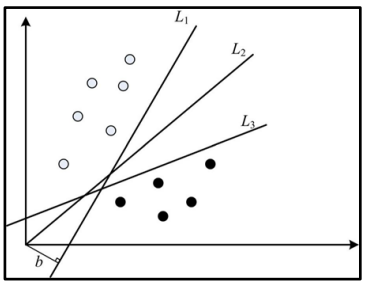

Рисунок 8.1 - Приклад класифікуючих розділяючих прямих

Випадок лінійної роздільності передбачає, що можливо побудувати таку пряму (площину), коли з кожного боку від неї будуть зібрані об'єкти лише одного класу. Звичайно, з точки зору практики, це майже неможливо, адже найчастіше знайдеться хоча б одна точка, що порушує це правило. Цей випадок відноситься вже до нелінійної роздільності.

Оскільки вибір роздільної гіперплощини нічим не обмежений, то цим можно скористатися для поліпшення класифікації. Розташуємо роздільну пряму так, щоб вона стояло максимально далеко від найближчих до неї точок обох класів.

Якщо формалізувати задачу розділення даних на два класи для просторів довільної розмірності: $\{ x_i , y_i\}$, $x_i \in \Re^d$, $y_i \in \{-1,1\}$, $i = 1,l$. Уявимо, що існує гіперплощина, *яка розділяє позитивні значення* ($y_i = 1$) і від'ємні ($y_i = -1$). Точки на гіперплощині задовільняють умові $wx + b = 0$

Для лінійно раздельного випадку метод опорних векторів шукає гіперплощину з максимальною величиною проміжку. Формально це означає, що вся навчальна вибірка задовольняє таким умовам (8.3 та 8.4):

$$wx_i \geq b + e \Rightarrow y_i = 1 \tag{8.3}$$

$$wx_i \leq b - e \Rightarrow y_i = -1 \tag{8.4}$$

Зауважимо, що параметри лінійного класифікатора визначені з точністю до нормування: алгоритм не зміниться, якщо w і b одночасно помножити на одну і ту ж позитивну константу.

Тоді після нормування маємо (8.5 та 8.6):

$$wx_i + b \geq 1 \tag{8.5}$$

$$wx_i + b \geq -1 \tag{8.6}$$

Ці умови можуть бути об'єднані в один набір нерівностей (8.7):

$$y_i (wx_i + b) - 1 \geq 0, \forall i . \tag{8.7}$$

У розглянутого методу є два суттєвих недоліки:

– метод не працює в разі, якщо класи лінійно нероздільні;

– припустимо, що в навчальній колекції є помилка – неправильно класифікований один або кілька елементів. Через ці елементи результативна лінія (рисунок 9.2) може сильно відрізнятися від тієї, яка вийшла б у випадку з коректною навчальної колекцією.

Переформулюємо постановку задачі з використанням функцій Лагранжу. Для цього є дві причини. По-перше, обмеження (8.7) будуть замінені обмеженнями у вигляді множників Лагранжа, які будуть набагато простіше в зверненні. По-друге, в такому формулюванні проблеми результат роботи алгоритму буде залежати не безпосередньо від даних навчальної вибірки, а від скалярних добутків векторів, що

входять в неї. Це дуже важлива властивість, яка дозволяє узагальнити метод опорних векторів у нелінійному випадку.

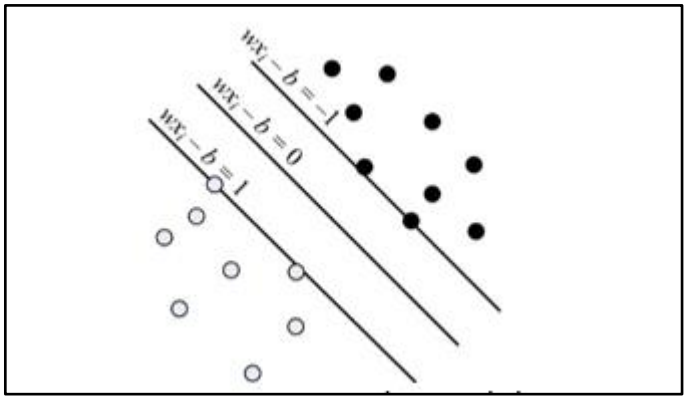

Рисунок 8.2 - Роздільна лінія

SVM це чорний ящик, який приймає на вхід характеристики даних, а на виході класифікацію за заздалегідь заданими категоріями. Як характеристику ми поставимо, наприклад, закінчення слова, а в якості категорій - частини мови.Щоб чорний ящик автоматично розпізнавав частина мови, для початку його потрібно навчити, тобто дати багато прикладів характеристик на вхід, і відповідні їм частини мови на вихід. SVM побудує модель, яка при достатніх даних буде коректно визначати частину мови.

При використанні такого підходу алгоритм має наступний вигляд.

Крок перший - читаємо файл корпусу і для кожного слова визначаємо його характеристики: саме слово, закінчення (2 і 3 останніх літери), приставка (2 і 3 перші літери), а також частини мови попередніх слів.

Крок другий - кожної частини мови та характеристиці присвоюємо порядковий номер і створюємо завдання для навчання SVM.

Крок третій - навчаємо модель SVM.

Крок четвертий - використовуємо навчену модель для визначення частини мови слів у реченні: для цього кожне слово потрібно знову подати у вигляді характеристик і подати на вхід SVM моделі, яка підбере найбільш відповідний клас, тобто частина мови.

9 ПРИЧИНИ ПОМИЛОК ПРИ РОЗПІЗНАВАННІ МОВИ ТА МОРФОЛОГІЧНОМУ АНАЛІЗІ

9.1 Акценти та шум

Один з очевидних недоліків розпізнавання мови - обробка акцентів і фонового шуму. Основна причина цього в тому, що велика частина тренувальних даних складається з американського говору з високим відношенням сигналу до шуму. Наприклад, в наборі розмов з телефонного комутатора є тільки розмови людей, чия рідна мова - англійська з невеликим фоновим шумом.

Але збільшення тренувальних даних саме по собі не розв'язувати цю проблему. Існує безліч мов, що містять багато діалектів і акцентів. Нереально зібрати розмічені дані для всіх випадків.

9.2 Семантичні помилки

Семантичні помилки - це помилки, пов'язані з невірним змістом дій і використанням недопустимих значень величин.

Пошук семантичних помилок набагато менш формалізований, ніж синтаксичних; частина їх з'являється при виконанні програми в порушеннях процесу автоматичних обчислень та індифікуються або видачею діагностичних повідомлень робочої програми, або відсутністю друку результатів із-за нескінченого повторення однієї і тієї ж частини програми, або появою непередбаченої форми чи змісту друку результатів

Часто кількість помилково розпізнаних слів не є самоціллю системи розпізнавання мови. Ми націлюємося на кількість семантичних помилок. Це та частка виразів, у яких ми неправильно розпізнаємо сенс.

Приклад семантичної помилки - коли хтось пропонує «let's meet up Tuesday» [давайте зустрінемося у вівторок], а розпізнавач видає «let's meet up today» [давайте

зустрінемося сьогодні]. Бувають і помилки в словах без семантичних помилок. Якщо розпізнавач не пізнав «up» та видав "let's meet Tuesday", семантика речення не змінилася.

9.3 Багато голосів в одному каналі

Розпізнавати записані телефонні розмови простіше тому, що кожного, хто говорить, записували на окремий мікрофон. Там не відбувається накладення декількох голосів в одному аудіоканалі. Люди ж можуть розуміти декількох ораторів, які іноді говорять одночасно.

Якісна система розпізнавання мови повинна вміти розділяти аудіопотік на сегменти залежно від мовця. Також вона повинна витягти сенс з аудіозаписи з двома голосами, які накладаються один на одного (поділ джерел). Це необхідно робити без мікрофона, розташованого прямо у рота кожного зі спікерів, тобто так, щоб розпізнавач працював добре, бувши розміщеним в довільному місці.

9.4 Якість запису

Акценти і фоновий шум - лише два фактори, до яких розпізнавач мови повинен бути стійким. Ось ще кілька:

– відлуння в різних акустичних умовах;

– артефакти, пов'язані з обладнанням;

– артефакти кодека, використовуваного для запису і стиснення сигналу;

– частота дискретизації;

– вік мовця.

Більшість людей не відрізнять на слух записів з mp3 і wav-файлів. Перш ніж заявляти про показники, які можна порівняти з людськими, розпізнавачі повинні стати стійкими і до перерахованих джерел варіацій.

9.5 Контекст

Ще одна причина помилок - розпізнавання без урахування контексту. У реальному житті ми використовуємо безліч різних додаткових ознак, які допомагають нам зрозуміти, що говорить інша людина. Деякі приклади контексту, використовувані людьми, і ігноровані розпізнавачами мови:

– історія бесіди і обговорювана тема;

– візуальні підказки про що говорить - виразу обличчя, рух губ;

– сукупність знань про людину, з яким ми говоримо.

Зараз у розпізнавачів мови в Android/iOS є список контактів, тому вони вміють розпізнавати імена друзів. Голосовий пошук на картах використовує геолокацію, щоб звузити кількість можливих варіантів, до яких ви хочете побудувати маршрут.

Точність систем розпізнавання збільшується з включенням в дані подібних сигналів.

9.6 Розгортання

Представляючи собі розгортання алгоритму розпізнавання мови, потрібно пам'ятати про затримки і обчислювальних потужностях. Ці параметри пов'язані, оскільки алгоритми, що збільшують вимоги до потужності, збільшують і затримку. Затримка - час від закінчення промови користувача і до закінчення отримання транскрипції. Невелика затримка - типове вимога для розпізнавання. Вона сильно впливає на відчуття користувача від роботи з продуктом. Часто зустрічається обмеження в десятки мілісекунд. Це може здатися занадто суворим, але згадайте, що видача розшифровки - це зазвичай перший крок в серії складних обчислень. Наприклад, в разі голосового інтернет-пошуку після розпізнавання мови потрібно ще встигнути виконати пошук.

Двонаправлені рекурентні шари - типовий приклад поліпшення, що погіршує ситуацію із затримкою. Всі останні результати розшифровки високої якості виходять з їх допомогою. Проблема тільки в тому, що ми не можемо нічого підраховувати після проходу першого двонаправленого шару до тих пір, поки людина не закінчив говорити. Тому затримка збільшується з довжиною речення.

На обчислювальну потужність впливають економічні обмеження. Необхідно враховувати вартість кожного поліпшення точності розпізнавача. Якщо поліпшення не досягає економічного порогу, розгорнути його не вийде.

Класичний приклад постійного поліпшення, яке ніколи не розгортають - спільне глибинне навчання. Зменшення кількості помилок на 1-2% рідко виправдовує збільшення обчислювальних потужностей у 2-8 разів.

9.7 Проблема поза-словникових слів

Існуючі системи розпізнавання мови містять моделі десятків і сотень тисяч слів, однак, як і при побудові моделей фонем, ніякі бази даних не можуть забезпечити повне покриття словника висловах реальної експлуатації. Зрозуміло, що якщо не передбачити способів обробки таких випадків, поза словникове слово, або OOV-слово (Out Of Vocabulary) буде розпізнано, як одне зі слів словника - IV-слово (IV - InVocabulary). Причому така вставка в текст може викликати ланцюжок додаткових помилок.

Слова не розпізнаються послідовно, одне за іншим - рішення відкладається до моменту розпізнавання останнього слова в ланцюжку, при цьому зберігається кілька варіантів ланцюжків (гіпотез). Те, що слово не включене в словник, означає, що його завжди апріорна ймовірність дорівнює нулю, і його участь в будь-який гіпотезі виключено – створення ймовірностей теж дорівнюватиме нулю. Замість OOV-слова буде підставлена якесь співзвучне слово, або декілька коротких IV-слів. Оскільки поєднання слів в моделях мови мають певні ймовірності, помилка може

поширитися на сусідні слова. Варто відзначити також, що неправильне розпізнавання OOV-слів може приводити до моделей з низькими можливостями, що призводить до необхідності збільшувати кількість розглянутих гіпотез, що, в свою чергу, збільшує обсяг обчислень. З огляду на те, що системи розпізнавання з великими словниками працюють на межі обчислювальних можливостей існуючих комп'ютерів, такий сценарій дуже небажаний.

Аналіз показує, що найбільшу частку серед OOV-слів займають нові терміни, імена, назви. Це як раз ті слова, які найчастіше визначають зміст висловлювання, тобто, власне, ті слова, заради яких фраза і була виголошена. Інакше кажучи, OOV-слова можуть нести великий обсяг інформації.

З вищесказаного випливає, що завдання обробки OOV-слів дуже важливе і повинне включати наступні підзадачі:

– визначення наявності та положення слова у словосполученні;

– розпізнавання послідовності фонетичних одиниць, складових слів;

– визначення написання слова.

Вирішувати проблему OOV-слів можна кількома способами, або їх комбінаціями:

– збільшення розміру словника;

– введення загальної моделі слова в словник - розширення ідеї моделей заповнення (Filler models), або моделей сміття (garbage) і моделей немовних звуків;

– використання системи з двома фазами розпізнавання, на першій з яких розпізнаються більші, ніж фонеми, одиниці - Sub-Word Units (Наприклад, склади, або отримані автоматично поєднання фонем);

– використання довірчих оцінок, отриманих різними системами розпізнавання.

10 РЕАЛІЗАЦІЯ ПРОТОТИПУ

10.1 Користувацький інтерфейс та взаємодія

Сьогодні всі призначені для користувача інтерфейси стають все більш мінімалістичним і простими. Дійсно, чим простіше інтерфейс, тим швидше і комфортніше буде користуватися вашим сервісом або додатком. І замість того, щоб пропонувати користувачеві складні формочки, в яких потрібно перемикатися між полями, щось набирати, десь щось вибирати і т.д., буває простіше і зручніше ввести кілька слів в одному полі або просто проговорити їх. Більш того, наприклад в Андроїд в будь-який момент можна натиснути на микрофончик і вимовити ті дані, які не хочеться / незручно / довго забивати руками. В iOS ситуація з голосовим уведенням теж покращилася в зв'язку з підтримкою російської в диктування. Тому було вирішено зробити мінімальний інтерфейс для взаємодії з користувачем.

Як видно із попередніх пунктів, то програма працює у такому руслі (рисунок 10.1):

– запис промови на телефоні;

– відправка запису на сервер;

– перетворення мови у текст;

– повернення отриманого результату для наступної обробки в json форматі;

– відображення отриманого результату користувачу;

– лексичний аналіз отриманого json файлу та підготовка даних до відправки на телефон у форматі команд;

– після отримання даних мобільний додаток зберігає підготовлені та оброблені дані до календаря;

– відображення отриманих результатів;

– обробка помилок.

Тобто програма складається з мобільного додатку та серверної частини для обробки даних.

Тому для кінцевого користувача не потрібно різноманіття форм. Було

вирішено залишити лише 1 кнопку для досягнення максимального комфорту під час використання додатку. Порівняння з аналогами системи також показало, що найбільш приємним є не наявність безлічі налаштувань та кастомізації, а лише легкість при користуванні. Так деякі аналоги взагалі відійшли від звичних андроїд додатків, а почали розробляти віджети та глибоко інтегровані системі в андроїд екосистему.

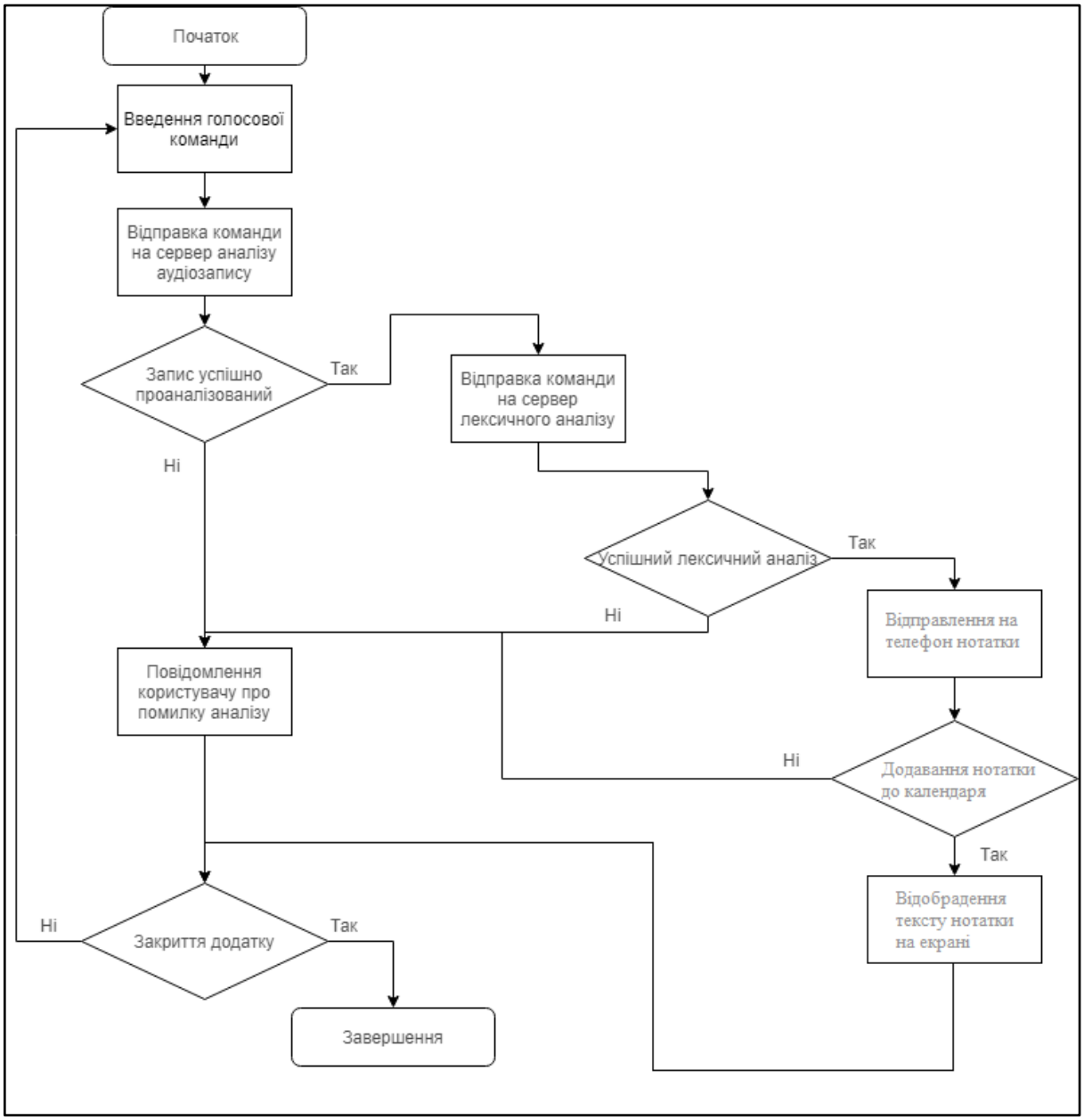

Рисунок 10.1 – Алгоритм роботи системи

Був розроблений mockup (рисунок 10.2), на базі якого потім будувався сам додаток.

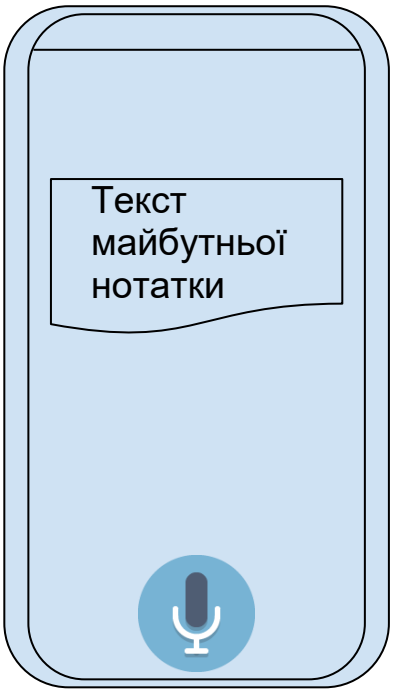

Рисунок 10.2 - Mockup ескіз додатку

Діаграма діяльності (англ. activity diagram) — в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Діаграма активності (рисунок 10.3) наведена для опису схема роботи прототипу програмної реалізації системи і використовується для моделювання виконання операцій візуалізації особливостей реалізації операцій класів та подання алгоритмів їх виконання.

Наведено графічне опис прототипу програмної реалізації з використанням мови об'єктного моделювання в області розробки програмного забезпечення UML (Universal Meta Language) як найбільш наочної візуалізації, проектування та документування.

Складність, продуктивність і час відгуку системи безпосередньо залежить від використовуваних їй в конкретний момент часу модулів і компонентів, а також від користувача і виду його діяльності.

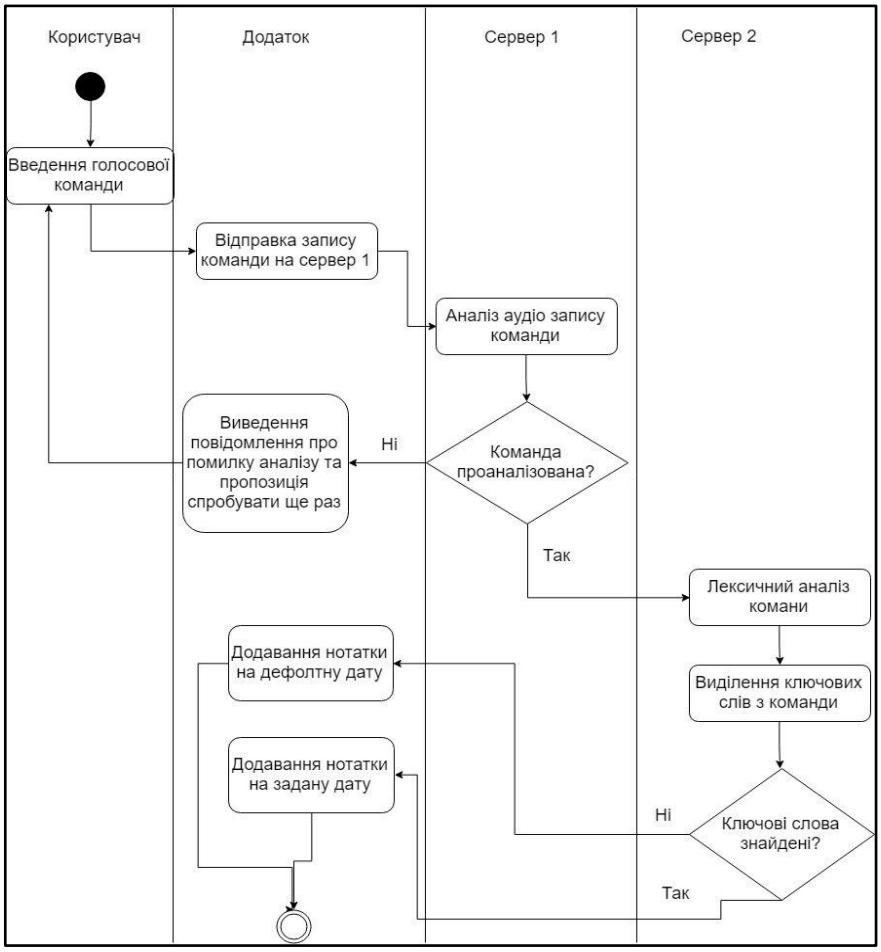

Рисунок 10.3 - Діаграма активності програмної реалізації системи

Use Case - це сценарна техніка опису взаємодії. За допомогою Use Case може бути описано і призначене для користувача вимога, і вимога до взаємодії систем, і опис взаємодії людей і компаній в реальному житті.

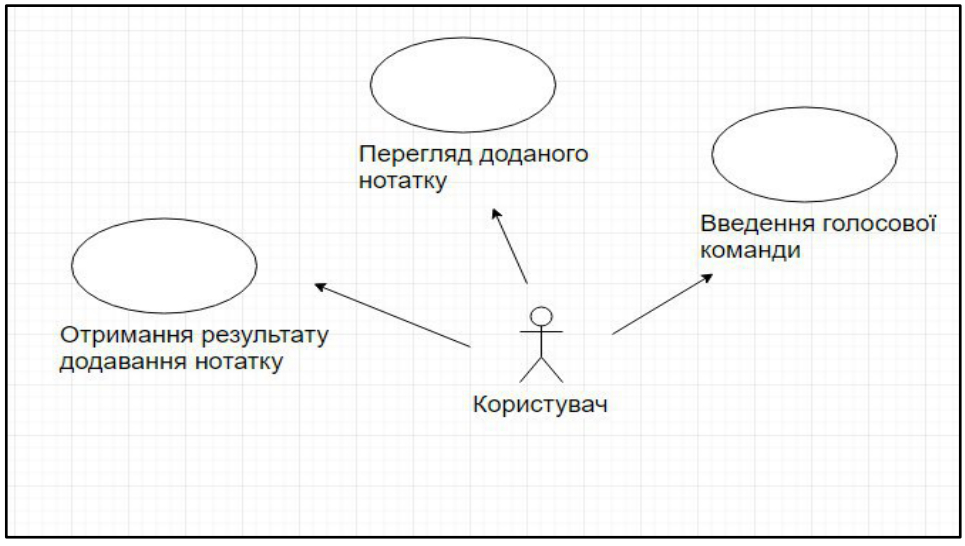

Рисунок 10.4 - Usecase діаграма системи

У загальному випадку, за допомогою Use Case може описуватися взаємодія двох або більшої кількості учасників, що має конкретну мету. Для аналізу була розроблена діаграма додатку (рисунку 10.4).

10.2 Розробка архітектури та проектування системи

Додаток реалізований на мобільному пристрої Android з вже вбудованою можливістю використання Google Speech Recognition API. Ми вирішили використовувати саме ці технології на основі проведених дослідів та отриманих результатів. Також ця система доволі стрімко розвивається, саме тому при виході нової версії, нам не потрібно буде виконувати складних кроків. Google API дуже комфортний у використанні, це також вплинуло на вибір технологій. Після перетворення мови за допомогою вище вказаного інтерфейсу отриманий результат надсилається до серверу шляхом HTTP запиту, де обробляється та повертає результат до мобільного пристрою (рисунок 10.5).

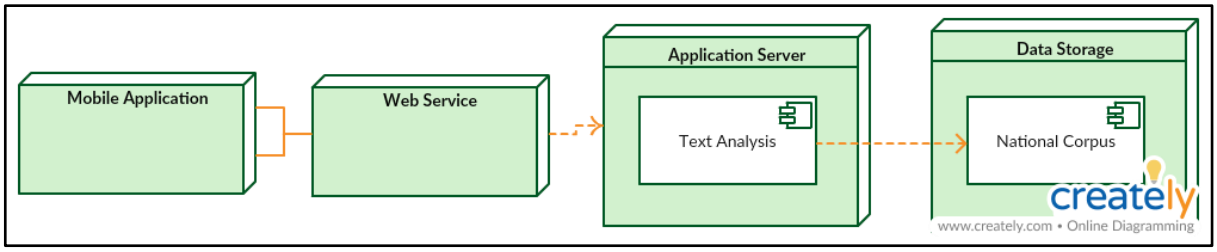

Рисунок 10.5 - Deployment діаграма

Нижче зображено діаграма аналізу документу(10.6).

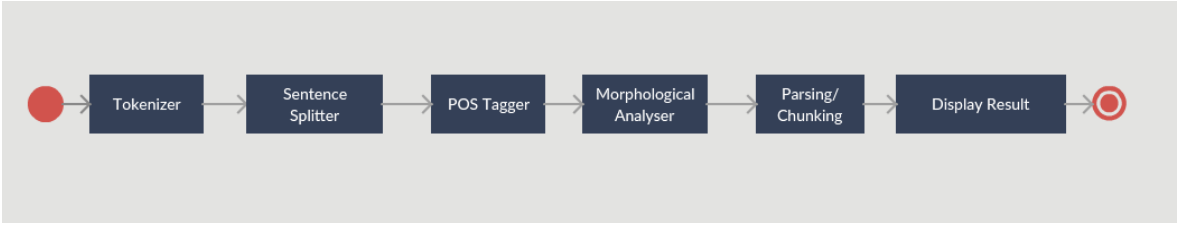

Рисунок 10.6 – Діаграма діяльності (activity diagram) аналізу документу

Для реалізація застосування було обано такі технології:

– Java для реалізації додатку на Android;

– Google Speech Recognition API;

– Python для реалізації морфологічного та синтаксичного аналізу;

– nltk, scikit-learn, pymorphy – open-source бібліотеки;

– Digitalocean – сервер для розгортання системи;

– unirest-java версії 1.3.1 - для надсилання REST запитів.

Нижче наведено приклад ініціалізації speech recognizer та RecognitionProgressView.

```
SpeechRecognizer speechRecognizer =
SpeechRecognizer.createSpeechRecognizer(context);
RecognitionProgressView recognitionProgressView =
(RecognitionProgressView) findViewById(R.id.recognition_view);
recognitionProgressView.setSpeechRecognizer(speechRecognizer);
recognitionProgressView.setRecognitionListener(new
RecognitionListenerAdapter() {        @Override
            public void onResults(Bundle results) {
                showResults(results);
            }
});
```

Коли SpeechRecognizer і RecognitionProgressView готові то можна використовувати speech recognizer як звичайно:

```
listen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {startRecognition();}
});
private void startRecognition() {
    Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
getPackageName());
```

```
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        speechRecognizer.startListening(intent);}
```

Нижче наведено приклад реалізації аналізу документу англійською мови.

```
import nltk
from nltk.corpus import state_union
from nltk.tokenize import PunktSentenceTokenizer
train_text = state_union.raw("training0.txt")
sample_text = state_union.raw("training1.txt")
custom_sent_tokenizer = PunktSentenceTokenizer(train_text)
tokenized = custom_sent_tokenizer.tokenize(sample_text)
def process_content():
    try:
        for i in tokenized[:5]:
            words = nltk.word_tokenize(i)
            tagged = nltk.pos_tag(words)
            print(tagged)
    except Exception as e:
        print(str(e))
process_content()
```

Вхідні дані:

Peace at home, peace on earth.

Результат:

[(‘Peace’, ‘NN’), (‘at’, ‘DT’), (‘home’, ‘NN’), (‘,’, ‘,’), (‘peace’, ‘NN’), (‘on’, ‘IN’), (‘earth’, ‘NN’), (‘.’, ‘.’)]

Приклад навчання для української мови, який читає файли корпусу за допомогою rnc.Reader, а потім викликає метод Tagger.train:

```
import sys
import re
import rnc
```

```
        import pos
        sentences = []
        sentences.extend(rnc.Reader().read('tmp/media1.xml'))
        sentences.extend(rnc.Reader().read('tmp/media2.xml'))
        sentences.extend(rnc.Reader().read('tmp/media3.xml'))
        re_pos=re.compile('([\w-]+)(?:[^\w-
]|$)'.format('|'.join(pos.tagset)))
        tagger = pos.Tagger()
        sentence_labels = []
        sentence_words = []
        for sentence in sentences:
            labels = []
            words = []
            for word in sentence:
                gr = word[1]['gr']
                m = re_pos.match(gr)
                if not m:
                    print(gr, file = sys.stderr)
                pos = m.group(1)
                if pos == 'ANUM':
                    pos = 'A-NUM'
                label = tagger.get_label_id(pos)
                if not label:
                    print(gr, file = sys.stderr)
                labels.append(label)
                body = word[0].replace('`', '')
                words.append(body)
            sentence_labels.append(labels)
            sentence_words.append(words)
        tagger.train(sentence_words, sentence_labels, True)
        tagger.train(sentence_words, sentence_labels)
        tagger.save('tmp/svm.model', 'tmp/ids.pickle')
```

Щодо реалізації другої частини додатку, а саме аналізу отриманого тексту ми розгорнули сервіс на сторонньому сервері Digital Ocean та спілкуємося з ним за

допомогою http-запитів. Для цього будуть надсилатись REST запити з отриманим словом на endpoint-и для отримання характеристик. Для надсилання REST запитів була використана бібліотека unirest-java версії 1.3.1. Приклад використання якої описано нижче.

```
HttpResponse<JsonNode>response=
Unirest.get("http://104.248.33.242:8080/morph-analysis")
.header("Accept","application/json")
.asJson();
```

Нижче наведено зовнішній вигляд програмного забезпечення від стартової сторінки до сторінки отримання кінцевого результату. На рисунку 10.7 зображено початкову сторінку застосування, щоб почати процес обробки голосового сигналу необхідно натиснути на зображення мікрофону.

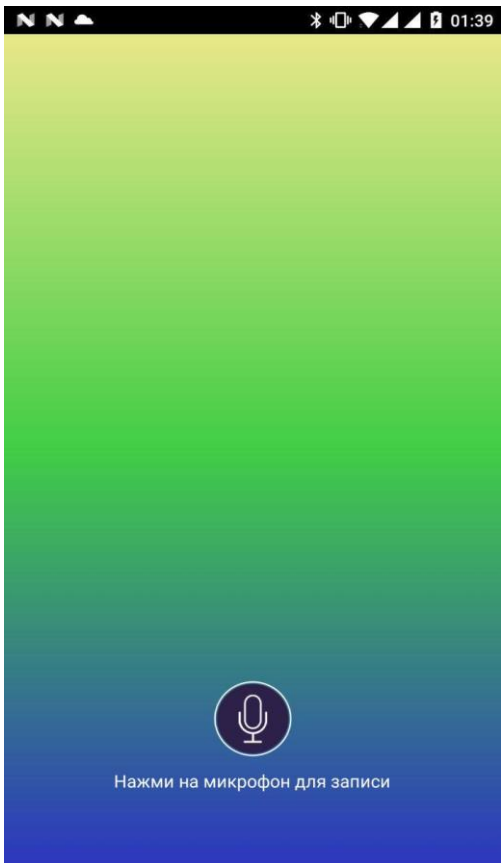

Рисунок 10.7 - Стартова сторінка додатку та сторінка розпізнавання мови

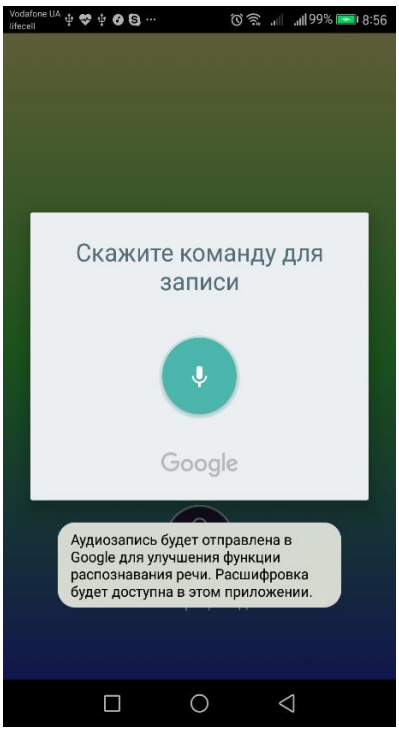

Рисунок 10.8 – Додаток у стані прослуховування команди

На рисунку 10.9 видно результат аналізу отриманого результату з серверу.

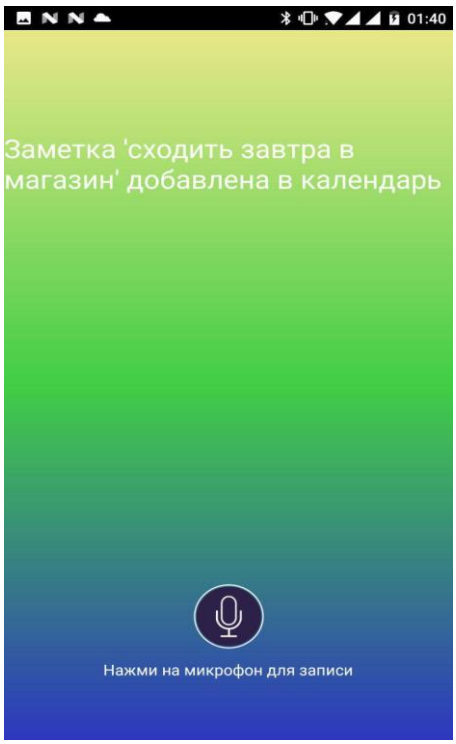

Рисунок 10.9 – Результат перетворення мови у текст

Результатом виконання команди може бути успішний чи не успішний аналіз голосового запиту (рисунок 10.10).

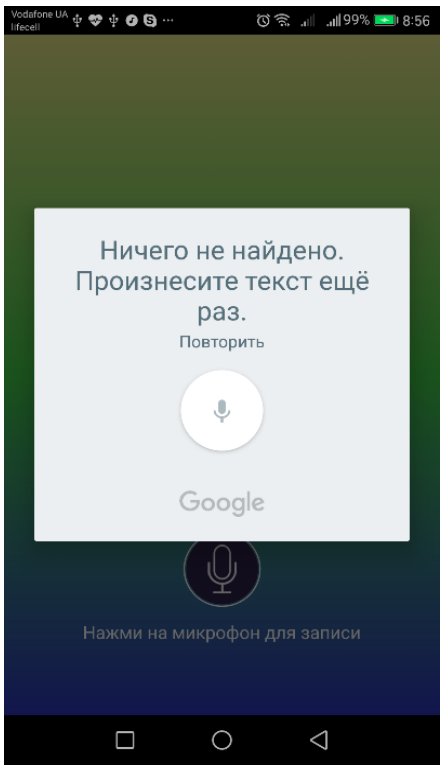

Рисунок 10.10 – Повідомлення про помилку аналізу команди

Розроблене застосування являє собою прототип для вивчення можливостей сервісів розпізнавання мови.

10.3 Тестування застосування

Тести дають впевненість, що програма працює як задумано. Тести можна запускати багато разів. Успішне виконання тестів покаже розробнику, що його зміни не зламали нічого, що ламати не планувалося.

Впавший тест дозволить виявити, що в коді зроблені зміни, які змінюють або ламають його поведінку. Дослідження помилки, яку видає впавший тест, і порівняння очікуваного результату з отриманим дасть можливість зрозуміти, де виникла помилка, будь вона в коді або у вимогах.

Тестування програмного забезпечення - перевірка відповідності між реальним і очікуваним поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином. У більш широкому сенсі, тестування - це одна з

технік контролю якості, що включає в себе активності з планування робіт (управління Test), проектування тестів (Test Design), виконання тестування (Test Execution) і аналізу отриманих результатів (аналіз Test).

При розробці застосування використовувалися unit-тести. При розробці було виділено Android-незалежні тести в окремий Java проект, їх запуск відбувався на JVM комп'ютера. Бібліотека Robolectric вирішила проблему швидкості запуску тесту. Robolectric дозволяє тестувати велику частину функціональності Android, включаючи layouts, GUI, сервіси, роботу з мережею, віджети. Також для unit-тестування використовувалися бібліотеки Junit та Mockito.

А також будо встановлено додаток на різних пристроях: використовувалися емулятори Android пристроїв, смартфон Xiomi Redmi 4 Pro, One Plus 3t та планшет Nexus 7. Всі вище вказані знімки екрану зроблені з телефону One Plus 3t.

ВИСНОВКИ

Дипломна робота присвячена актуальній темі аналізу сучасних інтерфейсів та фреймворків для розробки інтерфейсу перетворення мови у текст. Можна вважати виконананими завдання, які були поставленні перед початком виконання роботи.

Досліджено весь процесс перетворення мови у текст, починаючи з етапу промови мовця до отримання результату у текстовому вигляді. Розглянуто основні проблеми та особливості на кожному з етапів.

Досліджено характеристики та особливості використання існуючих інтерфейсів розпізнавання мови, серед яких: Google Speech Recognition API, Microsoft Speech API, CMUSphinx, NLTK.

Сформовано тестові дані для реалізації та проведення дослідів серед провідних сервісів Google Speech Recognition API, Amazon Alexa, Microsoft Speech API.

У дипломній роботі було визначено метрики та методи оцінювання систем аналізу мовлення. Знайдено та розглянуто причини помилок при розпізнаванні мови та засоби поліпшення результатів.

Було розроблено архітектуру та реалізовано застосування за допомогою використання наступних ключових технологій Java, Android API, Google Speech Recognition API, Python.

Було проведено тестування та аналіз роботи розробленого прототипу.

У майбутніх роботах можливе дослідження аналізу мовлення без звернення до сторонніх сервісів, а виконуючи обробку на робочому пристрої.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Somers, H. Machine Translation: Latest Developments. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003.

2. Маннинг К., Рагхаван П., Шютце Ч. Введение в информационный поиск — М.: Вильямс, 2011.

3. Васильев В. Г., Кривенко М. П. Методы автоматизированной обработки текстов. — М.: ИПИ РАН, 20084.

4. Барсегян А.А. и др. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP — 2-е изд. — СПб.: БХВ-Петербург, 2008.

5. Harabagiu, S., Moldovan D. Question Answering. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003.

6. Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце, Введение в информационный поиск // Вильямс - 2011.

7. Pang B. Opinion Mining and Sentiment Analysis. / B. Pang, L. Lee. – N.Y.: Now Publishers Inc., 2008.

8. Большаков, И.А. КроссЛексика — большой электронный словарь сочетаний и смысловых связей русских слов. // Комп. лингвистика и интеллект. технологии: Труды межд. Конф. «Диалог 2009». Вып. 8 (15) М.: РГГУ, 2009.

9. Bateman, J., Zock M. Natural Language Generation. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003.

10. Виноград Т. Программа, понимающая естественный язык — М.: Мир, 1976.

11. Касевич В.Б. Элементы общей лингвистики. — М.: Наука, 1977

12. McGurk H., MacDonald J., “Hearing Lips and Seeing Voices // Nature, 264, 1976.

13. Чистович Л.А. и др., «Руководство по физиологии. Физиология речи. Восприятие речи человеком», «Наука», Ленинград, 1976.

14. Карпов А.А., “Реализация автоматической системы многомодального распознавания речи по аудио- и видеоинформации” // Автоматика и телемеханика. 2014, Т. 75, № 12.

15. Вихованець І. Р., Городенська К. Г. Теоретична морфологія української мови [Текст] / І. Р. Вихованець (ред.). — К. : Університетське видавництво «Пульсари», 2004. — (Академічна граматика української мови). — ISBN 966-7671-60-7.

16. Daniel Jurafsky, James H. Martin. Speech and Language Processing Prentice Hall, 2008.

17. Transformation-Based Error-Driven Learning and Natural Language Processing [Електроний ресурс] - Режим доступу: http://acl.ldc.upenn.edu/J/J95/J95-4004.pdf. (Дата звернення: 16.10.2018).

18. Тестелец Я.Г. Введение в общий синтаксис. М.: РГГУ, 2001.

19. Levenshtein V. I. Binary codes capable of correcting deletions, insertions and reversals // Sov. Phys. Dokl. 1966. Vol. 6.

20. Khokhlov Y., Tomashenko N. Speech recognition performance evaluation for LVCSR system // Proc. of the 14th Intern. Conf. “Speech and Computer” SPECOM—2011, Kazan, Russia. 2011.

21. Kurimo M., Creutz M., Varjokallio M., Arsoy E., Saraclar M. Unsupervised segmentation of words into morphemes // Proc. Interspeech-2006, Pittsburgh, PA. 2006.

22. Karpov A., Kipyatkova I., Ronzhin A. Very large vocabulary ASR for spoken russian with syntactic and morphemic analysis // Proc. Interspeech-2011, Florence, Italy. 2011.

23. Morris A. C., Maier V., Green P. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition // Proc. Interspeech-2004, Jeju Island, Korea. 2004.

24. Vilar J. M. Efficient computation of confidence intervals for word error rates // Proc. ICASSP-2008, Las Vegas, NV. 2008.

25. Text Encoding Initiative [Електронний ресурс] – Режим доступу: http://www.tei-c.org/index.xml. (Дата звернення: 08.11.2018).