Висновки. Проведено аналіз новітніх технологій та розглянуто проблеми їх використання у межах предметної області розробки програмного забезпечення додатку «Тайм-менеджер» до операційної системи Android. За результатами проведених досліджень виконано аналіз структури та методів побудови програмних додатків для операційної системи Android.

За допомогою Intellij IDEA та Android Studio створено програмний додаток, який містить у собі наступний функціонал: механізм будильника для нагадувань про діяльність; механізм відстеження часу та таймерів з вказівкою типу обмеження; графічний механізм відображення витраченого часу.

При створенні програмного продукту вирішено проблеми перезавантаження пристрою, спрацювання таймеру під час сну телефону, відновлення роботи будильників після випадкового перезавантаження, багатопоточного запуску таймерів, обробки зміни орієнтації екрану, роботи у фоновому режимі.

ЛІТЕРАТУРА

1. Sovelluksen Valmistaja Create an elegantly designed Reminder/Alarm clock application [Электронный ресурс] / Learn Android Development | Download Free Apps. – 2014. – Режим доступу: http://www.appsrox.com/android/tutorials/remindme/.
2. Отзывчивое Android-приложение или 1001 способ загрузить картинку [Электронный ресурс] / Блог компании EastBanc Technologies, Разработка под Android. – 2013. – Режим доступу: http://habrahabr.ru/company/eastbanctech/blog/ 192998.
3. Сухоруков И. Многопоточность в Java / Программирование, Параллельное программирование, JAVA. – 2012. – Режим доступу: http://habrahabr.ru/post/164487.
4. Material Design: на Луну и обратно [Электронный ресурс] / Блог компании REDMADROBOT. – 2015. – Режим доступу: http://habrahabr.ru/company/redmadrobot/blog/252773/.

Надійшла до редколегії 26.04.2016.

004.9:004.912

ДРАНИШНИКОВ Л.В., д.т.н., професор
ШКУРКО О.А., магістр

Дніпродзержинський державний технічний університет

АНАЛІЗ ТА ВИЛУЧЕННЯ ІНФОРМАЦІЇ З ТЕКСТІВ

Вступ. Комп’ютерна лінгвістика (також відоме поняття "обробка природної мови" від дослівного перекладу з англійської naturallanguageprocessing) – це надзвичайно важлива область комп’ютерних наук яка ґрунтується на знаннях інформатики, лінгвістики і, все частіше, на теорії ймовірності та статистики.

На високому рівні абстракції комп’ютерна лінгвістика використовує комп’ютери при обробці людської (природної) мови (рис.1). З одного боку цієї проблеми ми маємо те, що часто називають розумінням природної мови, де ми беремо текст в якості вхідних даних для комп’ютера, а потім він обробляє текст і робить щось корисне. З іншого боку, ми маємо те, що часто називається naturallanguagegeneration, де комп’ютер, у деякому сенсі, виробляє мову у спілкуванні з людиною (користувачем системи).

Одна з найстаріших програм і проблема великої важливості – це машинний переклад. Це проблема зіставлення речень однією мовою з реченнями іншою мовою. І це дуже-дуже складне завдання. Помітний швидкий прогрес в останні 10-20 років у цій області. Наприклад, подивимось на результат перекладу фрази, яка багатьом знайома, Googletranslate з української на англійську. Хоча переклад далекий від ідеалу, ви все одно зможете зрозуміти багато з того, що було сказано в оригіналі.

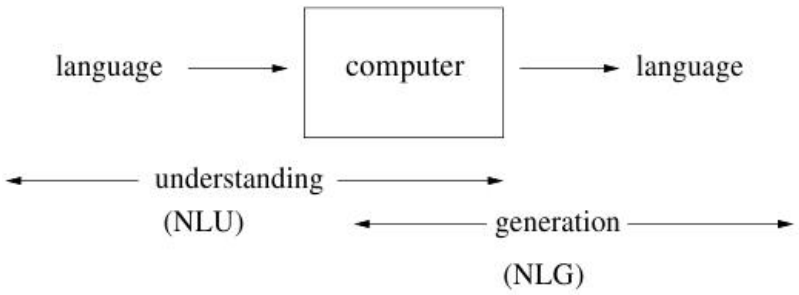

Рисунок 1 – Комп'ютерна лінгвістика як абстрактна система вводу/виводу

Другий приклад застосування – те, що часто називають вилученням інформації (information-extraction). Мета в даному випадку така: взяти якийсь текст у якості вхідних даних і створювати деякі структуровані, тобто представлені у вигляді бази даних, представлення деяких ключових частин тексту. Розглянемо це застосування на прикладі пошуку вакансій. На вході ми маємо вакансію. Вихід охоплює різні важливі аспекти цієї вакансії, наприклад, галузь, посаду, місто, компанію, зарплату, і т.і. Виходить, що ця інформація є витягом з цього документа. Ці важливі для витягу аспекти представлені у вигляді частини природної мови. У цьому і полягає проблема важливості розуміння природної мови. В якомусь сенсі це просто перетворення неструктурованого тексту на вході до структурованого представлення інформації у вигляді бази даних. Тут можна чітко сформулювати мету вирішення цієї конкретної проблеми – вилучення інформації. Після того, як ми виконали цей крок, ми можемо, наприклад, виконувати складний пошук. Наприклад, необхідно знайти всі робочі місця в рекламній індустрії з конкретним розміром заробітної платні. Такий пошук дуже важко сформулювати за допомогою звичайної пошукової системи, але при першому запуску система отримання інформації через веб-сайти видає всі вакансії, які знаходяться в інтернеті. Потім можна виконати запит до бази даних і виконувати набагато більш складні запити, ніж цей. Крім того, можна б було виконати і статистичні запити. Можливо було б запитати, яка кількість робочих місць у бухгалтерії змінилась за ці роки або кількість робочих місць в області розробки програмного забезпечення в Києві за останній рік.

Інше ключове застосування комп'ютерної лінгвістики – це анотування текстів. У такому випадку вирішується задача витягу фактів для створення анотації з одного або декількох документів. Один з прикладів реалізації алгоритмів анотування текстів – система NewsBlaster. В якості вхідної інформації вона сприймає досить багато документів про певну новину, а на виході видає анотацію до всіх новин певної тематики. Використання такої системи корисне для обробки великих масивів даних з метою вилучення ключової інформації.

Великий обсяг накопиченої інформації і висока швидкість надходження нової пред'являють все більш жорсткі вимоги до сучасних інформаційних порталів. По-перше, за умови, що масиви даних постійно розростаються, стає важко (практично неможливо) знайти потрібну інформацію; по-друге, дані часто дублюються і суперечать одне одному. Для вирішення цих проблем необхідний перехід на новий якісний рівень при обробці інформації – необхідно вести обробку на семантичному рівні, тобто враховувати зміст документів, що надходять. Така обробка тексту природною мовою забезпечується системами автоматичного аналізу, що використовують лінгвістичний підхід.

Постановка задачі. Окрім описаних вище застосувань комп'ютерної лінгвістики розглянемо деякі базові проблеми цієї області науки, які залежать від цих застосувань.

Перша проблема – це те, що називається проблемою маркування. На абстрактному рівні проблема виглядає так: в якості вхідної інформації маємо деяку послідовність символів, на виході ми повинні отримати таку ж послідовність, але кожен символ має бути співвіднесений з певною категорією. Розглянемо декілька прикладів маркування текстів. Перший – розпізнання частин мови. Як вхідну інформацію система

сприймає речення, на виході кожне слово речення промарковане залежно від частини мови. Це одна з основних задач комп'ютерної лінгвістики. Якщо вирішувати її з достатньо великою точністю, вона буде дуже корисною для застосування в широкому колі інших застосувань. Другий приклад – розпізнання іменованих сутностей. Основа прикладу дуже подібна до попереднього прикладу, на вході є певне речення, на виході система маркує певні сутності, якщо вони присутні у реченні. Найчастіше маркують такі сутності: назви компаній, географічні назви, імена людей та посади. Основною метою роботи є розробка методу й алгоритму отримання даних з текстів новин у галузі вилучення даних.

Результати роботи. *Первинна обробка тексту.* Отже, на вході у нас текст природною мовою. Аналізувати його необхідно відразу на всіх лінгвістичних рівнях: графематичному, лексичному, морфологічному, синтаксичному, семантичному.

Текст ділиться на абзаци, речення, слова. Потім слова нормалізуються – виділяється їх початкова форма. Далі проводиться повний або частковий синтаксичний розбір, визначаються залежності і зв'язки між словами в реченнях.

На перший погляд здається, що розбити текст на речення не складає ніяких труднощів. Потрібно просто орієнтуватися на знаки пунктуації, що маркують кінець речення. Але працює цей метод далеко не завжди. Адже, наприклад, крапка може означати і скорочення, використовуватися в дробових числах або URL посиланнях. Будь-який знак може використовуватися у назвах компаній або сервісів, наприклад, Yahoo! або Yandex.Maps.

З виділенням початкової форми теж не все так просто. Звичайно, в більшості випадків її можна отримати за допомогою морфологічного словника. Але його можна застосовувати далеко не завжди. Наприклад, деякі слова можуть бути як іменником, так і дієсловом. І перш, ніж звертатися до морфологічного словника, потрібно зняти омонімію. Вирішується ця проблема за допомогою корпусу мови, у якому всі слова розмічені за частинами мови, і омонімія знята. Таким чином, ґрунтуючись на контексті і статистиці вживання слова в корпусі, можна прийняти рішення, до якої ж частини мови належить те чи інше слово.

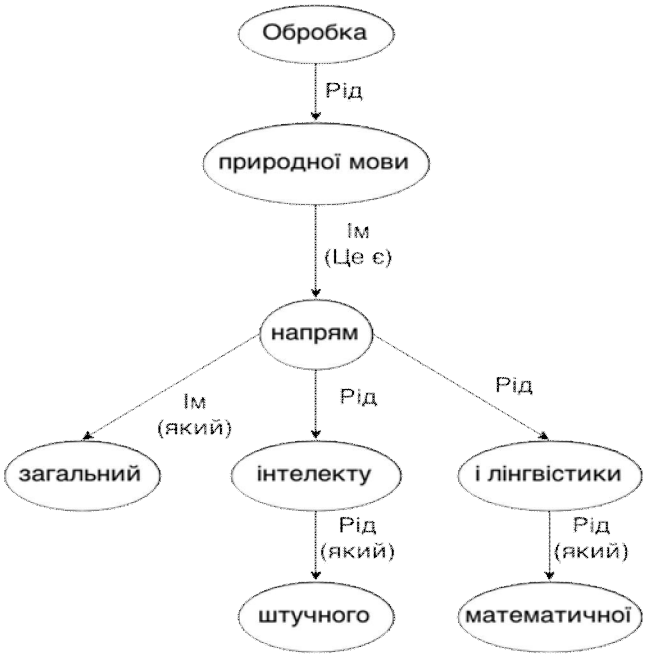

Рисунок 2 – Синтаксичне дерево як граф залежностей та відношень між словами речення, побудований за допомогою синтаксичного парсера

Наступний етап – повний або частковий синтаксичний розбір. Вибудовується граф залежностей і відносин між словами в межах речення. Ось приклад синтаксичного дерева, яке можна побудувати за допомогою синтаксичного парсера (рис.2).

Алгоритмів, які в будь-яких умовах можуть побудувати повний синтаксичний граф без помилок не існує. Однак для більшості прикладних задач TextMining достатньо і часткового розбору.

Крім морфологічної омонімії, про яку ми говорили вище, буває також синтаксична і «об'єктна» омонімія. В якості прикладу синтаксичної омонімії наведемо: «Тінь яблуні не заважає». Таке речення може бути інтерпретоване як «Тінь від яблуні не заважає» або як «Тінь

не заважає яблуні», і обидва варіанти прочитання будуть синтаксично правильні. Для вирішення такої проблеми потрібно залучати вже семантичний аналіз.

«Об'єктна» омонімія передбачає, що у двох різних реальних об'єктів можуть бути однакові найменування. Наприклад, в Україні є одразу декілька відомих людей з прізвищем Шевченко: письменник, футболіст та політик. Якщо не навчити систему розрізняти цих людей, при спробі вилучення фактів про них можуть виникати різноманітні казуси.

Витяг фактів. Коли всі ці кроки пройдені, можна переходити безпосередньо до отримання фактів. За допомогою спеціальних алгоритмів ми хочемо отримати з неструктурованого уривка тексту, в якому всі потрібні нам об'єкти та факти будуть розмічені і категоровані. Наочно уявити це можна наступним чином (рис.3).

POST	***Міністр економічного розвитку й торгівлі Айварас***	
FIO	***Абромавичус*** прогнозує приплив інвестицій в ***українську***	GEO
	економіку після закінчення війни на ***Донбасі***.	GEO
	Про це він заявив під час телеефіру на ***ТРК "Лтава"***.	COMP
	Він закликав позитивний приклад приходу як інвестор	
GEO COMP	***американської*** компанії ***Monsanto***, що вже вклала ***250 млн***	NUMBER
	доларів у будівництво заводу з виробництва насіння	
GEO	кукурудзи в ***Житомирі***. Серед негативних факторів міністр	
	зазначив падіння гривні.	
	З його слів, при сприятливих умовах поліпшення в ***українській***	GEO
	економіці можуть з'явитися вже в ***2016 році***.	DATE

Рисунок 3 – Текст з розміченими та категорованими сутностями

Умовно можна виділити три основні підходи, що застосовуються у витягу фактів: за онтологіями; спираючись на правила (Rule-based); спираючись на машинне навчання (ML – MachineLearning).

У нашому випадку онтології – це «концептуальні словники», що представляють собою структури, в яких описуються деякі поняття та/або об'єкти, відносини між ними, а також їх характеристики (рис.4).

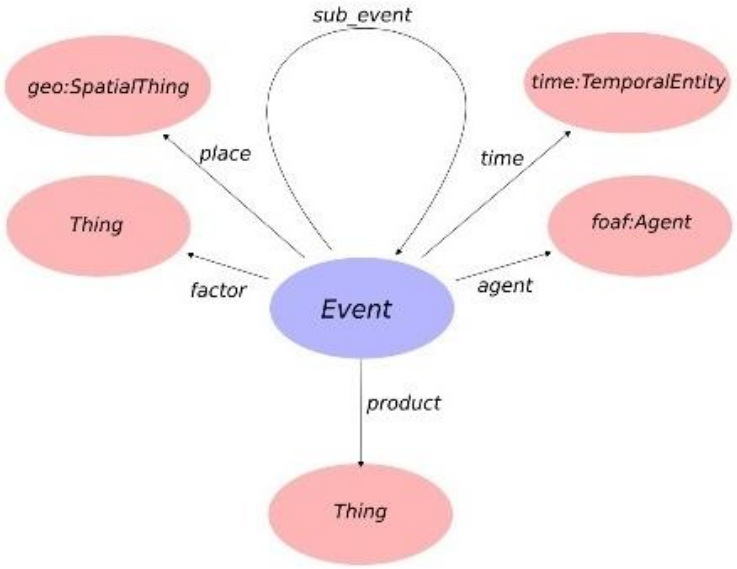

Рисунок 4 – Онтологія як концептуальний словник, що описує деякі поняття, об'єкти, відношення між ними та їх характеристики

Онтології можуть бути універсальними (в них робиться спроба описати максимально широкий набір об'єктів), галузеві (з інформацією за предметними областями) і вузькоспеціалізовані (призначені для вирішення конкретної задачі). Також можуть застосовуватися онтології об'єктів (бази знань). Найбільш яскравий приклад бази знань – це Вікіпедія.

Отже, у нас є певна онтологія. Спираючись на контексти і вже наявні списки об'єктів, можна будувати гіпотези по відношенню до об'єктів і фактів у тексті, а

далі верифікувати або відхиляти ці гіпотези. Зліва на рис.4 наведено текст, в якому кольором розмічені об'єкти, про які хотілося б отримати якусь інформацію. Як онтології застосуємо Вікіпедію (рис.5). Відправляючи туди запити за ключовими словами з нашого тексту, ми отримаємо список статей, розташований праворуч на рис.5. Червоним на рис 4 в ньому позначені статті, що відносяться відразу до кількох об'єктів.

Канцлер Німеччини	http://ua.wikipedia.org/wiki/Канцлер
Ангела Меркель	.../wiki/Канцлер_Німеччини
вважає, що суть	.../wiki/Німеччина
Мінської угоди	.../wiki/Ангела_Меркель
полягає у проведенні	.../wiki/суть
в Донецькій та	.../wiki/Мінська_угода
Луганській областях	.../wiki/Мінськ
виборів, згідно з	.../wiki/угода
українською	.../wiki/Донецька_область
Конституцією.	.../wiki/Луганська_область
	.../wiki/вибори
	.../wiki/Конституція
	.../wiki/Конституція_України

Рисунок 5 – Вікіпедія як онтологія

Тепер наша мета – відсіяти невірні гіпотези. Зробити це можна різними способами. Найчастіше застосовується машинне навчання, різні контекстні та синтаксичні фактори.

Витяг інформації за допомогою онтологій дозволяє отримати досить високу точність NER і відсутність випадкових спрацьовувань. Зняття омонімії також відбувається з високою точністю. До недоліків цього підходу можна віднести низьку повноту, адже отримати можна тільки те, що вже є в онтології. А в онтологію потрібно або додавати об'єкти руками, або вибудовувати процедуру автоматичного додавання.

Інший підхід – машинне навчання – вимагає великого обсягу ввідних даних. Потрібно максимально покрити лінгвістичною інформацією навчальну вибірку текстів: розмітити всю морфологію, синтаксис, семантику, онтологічні зв'язки. Плюси цього підходу в тому, що він не вимагає ручної праці, крім створення розміченого корпусу. Не потрібно складати правила або онтології. За необхідності така система легко перенастроюється і перенавчається. Правила виходять більш абстрактними. Однак є і мінуси. Інструменти для автоматичної розмітки українськомовних текстів поки не дуже розвинені, а існуючі – не завжди легко доступні. Корпуси повинні бути досить об'ємними, розмічені правильно, послідовно та повністю. А це досить трудомісткий процес. Крім того, якщо щось пішло не так, складно відстежити, де саме виникла помилка, і точково її виправити.

Третій підхід – підхід, заснований на правилах, тобто написання шаблонів уручну. Аналітик складає описи типів інформації, які потрібно отримати. Підхід зручний тим, що якщо в результатах аналізу виявляються помилки, дуже просто знайти їх причину і ввести необхідні зміни в правила. Найпростіше складаються правила для відносно стандартизованих об'єктів: імен, дат, назв компаній і т.і.

Вибір оптимального підходу визначається конкретною задачею. Зараз найчастіше застосовуються онтології і машинне навчання, однак майбутнє – за гібридними системами.

Для програмної реалізації була обрана задача оцінки тональності відгуків. Мета реалізації програмного засобу – створення інструменту, який буде з високою точністю

автоматично класифікувати відгуки, залишені у соціальній мережі Twitter за тональністю. В якості вхідної інформації система повинна сприймати слово або словосполучення. Під час роботи система відстежує та аналізує всі повідомлення з вхідним запитом у режимі реального часу. Як вихідну інформацію система видає знайдені повідомлення, результат сентиментального аналізу та мету даних повідомлень. Програмний засіб реалізовано мовою програмування Python з використанням API соціальної мережі Twitter та платформи NLTK. Власне алгоритм обробки відгуків базується на наївному баєсівському класифікаторі.

Класифікатор повинен пройти навчання, щоб вміти відрізняти позитивні, негативні та нейтральні відгуки. Для цього зберігається та маркується деякий набір речень (повідомлень соціальної мережі). Після цього система навчається, маркуючи тестові повідомлення.

Послідовність роботи системи та взаємозв'язки описаних вище функцій зображені на рис.6.

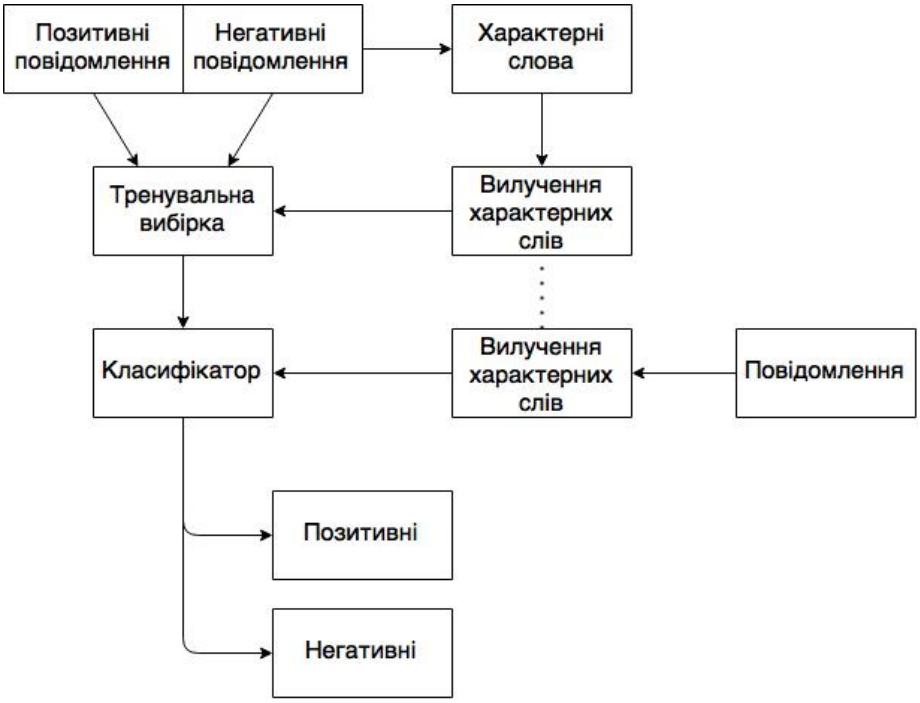

Рисунок 6 – Послідовність роботи та взаємозв'язки компонентів системи

Висновки. Розроблений програмний засіб може бути використаний для загального аналізу тональності відгуків. Але його можна адаптувати під конкретні потреби певної предметної області, наприклад, реалізувати вилучення та аналіз інформації про такі сутності, як ціна, якість, зовнішній вигляд або технічні характеристики. Також можна додати функцію вилучення іменованих сутностей, наприклад, таких як імена та прізвища, посади, назви організацій, дати або географічні назви.

ЛІТЕРАТУРА

1. Bird S. Natural Language Processing with Python. Analyzing Text with the Natural Language Toolkit / S.Bird, E.Klein, E.Loper. – O'ReillyMedia, 2009. – 504c.
2. Шумейко А.А. Интеллектуальный анализ данных (Введение в DataMining) / А.А.Шумейко, С.Л.Сотник. – Дн-вск: Белая Е.А., 2012. – 210с.

Надійшла до редколегії 26.04.2016.