МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КУРСОВА РОБОТА

з дисципліни «**Методи та системи штучного інтелекту**»

Тема «**Інтерактивна система штучного інтелекту – суперкомп'ютер WATSON**»

Виконали:
студенти 3 курсу
групи КС-33
Рузудженк С.Р.,
Ковальова Т.О.
Перевірив:
д-р фіз.-мат. наук, проф. Куклін В.М.

Харків – 2020

РЕФЕРАТ

Пояснювальна записка містить 24 сторінки, 8 джерел інформації.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, КОГНІТИВНА СИСТЕМА, IBM WATSON, СУПЕРКОМПЬЮТЕР, КОНТЕНТНА АНАЛІТИКА.

Робота присвячена дослідженню принципів роботи технології DeepQA, особливостей її архітектури, а також характеристиці роботи когнітивної системи штучного інтелекту IBM Watson, розробленої у рамках проекту DeepQA. Розглянута конструкція, а також програмна частина даної системи.

Результатом роботи стало отримання основних відомостей та розуміння механізмів роботи системи на основі технології DeepQA, а також процесу виявлення системою відповіді на поставлене запитання.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ШІ – штучний інтелект

ПЗ – програмне забезпечення

ОС – операційна система

QA (англ. Question Answering) – програмні системи, які надають прямі відповіді на питання, поставлені природною мовою

OAQA (англ. Open Advancement of Question Answering) – відкрите просування систем відповідей на запитання

IBM Cloud – сукупність хмарних технологій і сервісів компанії IBM

UIMA (англ. Unstructured Information Management Architecture) – архітектура для створення масштабованих додатків, які аналізують та вилучають інформацію з неструктурованих джерел даних, таких як текст, аудіо та відео

NLP (англ. Natural Language Processing) – обробка природної мови

ODSE (Ontology-Driven Software Engineering) – різноманітні методи застосування онтологій у процесі розробки ПЗ

ВСТУП

Когнітивні технології - це напрям розвитку систем штучного інтелекту, основне завдання яких - допомагати людині в прийнятті рішень у складних умовах. Існує ряд галузей і процесів, що вимагають управління при величезній кількості змінних параметрів, складних залежностей та результатів, які важко передбачити. При цьому рішення повинні прийматися в режимі часу, близькому до реального.

З цього випливає розуміння спектру клієнтів і можливих проектів, для яких підходять когнітивні технології: це в першу чергу великі компанії з тих галузей бізнесу, де необхідно швидко і якісно обробляти великі обсяги даних, виявляти складні зв'язки і залежності, за алгоритмами, близькими до людського мислення, генерувати варіанти рішень. Також це ті області, де потрібне освоєння і розуміння великої кількості інформації про нові розробки, облік накопиченого досвіду і високі ризики при прийнятті рішень. У таких галузях когнітивні технології особливо цінні.

Серед діючих когнітивних систем найбільш відомий – суперкомп'ютер IBM Watson з програмою штучного інтелекту, створеної під керівництвом Девіда Феруччі в рамках проекту DeepQA.

У даній курсовій роботі будуть розглянуті принципи роботи суперкомп'ютеру Watson, що базуються на технології DeepQA, його конструкція та апаратна частина, а також охарактеризовано процес формування системою відповіді на поставлене питання.

1 СТИСЛА ХАРАКТЕРИСТИКА РОЗГЛЯНУТИХ ТЕХНОЛОГІЙ

У даному розділі наведена стисла інформація про технології, системи, програмні підходи та типи архітектури, що розглядаються у ході даної курсової роботи.

ODSE (Ontology-Driven Software Engineering) – онтологічний підхід, що передбачає використання онтологій у процесі розробки ПЗ. Одним з формальних визначень [1] онтології є:

$$O = < C, R, F >, \qquad (1.1)$$

де C - кінцева множина концептів (понять) предметної області;

R – кінцева множина відносин між концептами;

F - кінцева множина функцій інтерпретації, заданих на концептах та/або відносинах.

Онтології використовують зарезервований словник термінів для визначення концептів відносин між ними для конкретної предметної області. За допомогою онтологій можна автоматизувати обробку семантики даних з метою її ефективного використання (перетворення, пошуку). Відповідний принцип обробки даних базується на представленні опису предметної області як бази знань, що містить поняття та взаємозв'язки, а також орієнтований на автоматизовану інтерпретацію і обробку інформації. Застосування онтологій в інформаційних системах дозволяє відобразити реальну картину світу у вигляді понять, відносин, а також виконувати різну інтерпретацію [2].

Класифікація питань (Question Classification) – це завдання виявлення типів питань або частин питань, які потребують спеціальної обробки. Це може включати в себе все, що завгодно, від одиничних слів з потенційно подвійним значенням до цілих клаузул, які мають певну синтаксичну, семантичну або риторичну функціональність, яка може інформувати наступні компоненти при їх аналізі.

Виявлення зв’язків (Relation Detection)

Компоненти виявлення зв’язків шукають смислові відносини в тексті. Це важливо, якщо різні терміни мають таке значення і корисно при відображенні відносин між іменниками або сутностями в питанні.

Декомпозиція (Decomposition)

Правильність інтерпретації питання й отриманої відповіді будуть вище після того, як будуть розглянуті всі зібрані докази і всі відповідні алгоритми. Навіть якщо питання не потрібно розкладати, щоб визначити відповідь, цей метод може допомогти підвищити загальну достовірність відповіді.

DeepQA вирішує паралельні питання, які можна розкласти, застосовуючи наскрізну систему контролю якості до кожного підрозділу і синтезуючи остаточні відповіді за допомогою компонента комбінації відповідей.

Генерування відповідей-кандидатів (Candidate Answer Generation)

Результати аналізу питання надходять до генерації кандидатів, де різні техніки, відповідно до класифікації питання, застосовуються для генерації відповідей-кандидатів. Система може генерувати кілька варіантів відповіді-кандидата на основі аналізу підрядків або аналізу посилань (якщо покладене в основу джерело містить гіперпосилання).

Якщо правильну відповідь на даному етапі не згенеровано у якості кандидата, система не зможе дати вірної відповіді на питання. Таким чином, цей крок значно сприяє більш точному відкликанню, оскільки очікується, що інша частина системи обробки видаватиме правильну відповідь, навіть якщо набір кандидатів досить великий. Тому однією з цілей проектування системи є допуск шуму на ранніх стадіях конвеєра і підвищення точності на виході.

Бінарний пошук (Primary search)

Мета бінарного пошуку полягає в тому, щоб знайти якомога більше інформації, що потенційно містить відповідь, на основі результатів аналізу питань. Основна увага приділяється відкликанню з очікуванням того, що інструменти більш глибокої контентної аналітики вилучать кандидатів та проведуть оцінку змісту, а також усіх доказів, які можуть бути знайдені в підтримку або спростування кандидатів, щоб підвищити точність.

М’яка фільтрація (Soft filtering)

До великого початкового набору відповідей-кандидатів застосовуються менш ресурсоэмны алгоритми оцінки, щоб звести його до мінімуму кандидатів для роботи з більш інтенсивними компонентами оцінки.

Hypothesis and Evidence Scoring

Для того щоб краще оцінити кожну відповідь- кандидата, яка пройшла м'яку фільтрацію, система збирає додаткові підтверджуючі докази. Архітектура підтримує інтеграцію різних методів збору доказів. Однією особливо ефективною технікою є пошук , де відповідь-кандидат додається в якості обов'язкової умови до основного пошукового запиту, що був сформований на основі даного питання.

Алгоритми підрахунку (scoring) визначають ступінь впевненості у тому, що знайдені докази підтверджують відповідь-кандидата. Структура DeepQA підтримує включення безлічі різних компонентів, які враховують різні докази та надають оцінку, яка відповідає тому, наскільки переконливими є докази, що підтверджують відповідь-кандидата на конкретний питання.

Final merging and ranking

Декілька відповідей - кандидатів можуть бути семантично еквівалентними, не дивлячись на різні форми представлення. Це може стати на заваді техніки ранжування, яка використовує відносні відмінності між кандидатами. Без проведення процесу злиття кандидатів, алгоритми оцінки будуть порівнювати різні форми, що представляють один відповідь. Для уникнення таких ситуацій використовується підхід, який використовує сукупність алгоритмів співставлення та нормалізації, для виділення еквівалентних і пов’язаних гіпотез, після чого проводить злиття та підрахунок балів. Після злиття система повинна ранжувати гіпотези і оцінити достовірність на основі їх об'єднаних балів.

Питально-відповідна (довідкова) система (QA System) – це інформаційно-пошукова система, в якій у відповідь на поданий запит очікується пряму відповідь, а не набір посилань, які можуть містити відповіді. Така система є гібридом пошукових та інтелектуальних систем (часто вони розглядаються як інтелектуальні пошукові системи). QA-система повинна бути здатна приймати

питання природною мовою. Інформація надається на основі документів з мережі Інтернет або з локального сховища.

Архітектура управління неструктурованою інформацією (UIMA) – архітектура, що підтримує технологію управління неструктурованою інформацією, яка дозволяє аналізувати та отримувати знання з неструктурованої інформації.

UIMA Асинхронне масштабування (UIMA-AS) – це набір можливостей, який надає більш гнучкі і потужні можливості масштабування і розширює підтримку компонентів UIMA.

UIM-додаток – додаток, що дозволяє отримувати знання з неструктурованої інформації, у тому числі з текстів, аудіо, відео та зображень.

Обробка природної мови (NLP) – підрозділ інформатики та ШІ, присвячений тому, як комп'ютери аналізують природні мови. NLP дозволяє застосовувати алгоритми машинного навчання для тексту й мови.

Захоплення даних (Acquisition) – забезпечує збір документів з різних джерел і формування необхідних колекцій, призначених для конкретних програм.

Web-павуки (web crawler) – програма, що є складовою частиною пошукової системи та призначена для обходу сторінок інтернету з метою занесення інформації про них (ключових слів) до бази даних.

Інтерфейс Collection Reader – інтерфейс, який зв'язує додатки з колекціями даних і метаданих.

Аналіз неструктурованої інформації (Unstructured Information Analysis)

Поділяється на два послідовних етапи – спочатку виконується аналіз документів, а потім аналіз колекцій документів.

Текстові аналітичні машини (Text Analysis Engine) – різноманітні транслятори та модулі, які виконують граматичний розбір, класифікацію, узагальнення. Використовуючи вхідні документи, текстові аналітичні машини виробляють узагальнені аналітичні структури.

Узагальнені аналітичні структури (Common Analysis Structure) – об'єктно-орієнтована структура даних, що логічно містить документи для аналізу. За допомогою CAS можна представляти об'єкти, їх властивості та значення.

Аналіз на рівні колекцій (Collection Level Analysis)

На етап аналізу колекцій документи можуть надходити безпосередньо або через проміжний етап, на якому виконується необхідна фільтрація та переформатування для подальшої паралельної обробки. Аналіз на рівні колекцій дозволяє узагальнити відомості, що містяться в колекції документів.

Аналіз структурованої інформації (Structured Information Analysis)

Використовується як для вхідних даних, що надходять в структурованій формі, так і для даних, що з'являються після аналізу неструктурованої інформації, де їх значна частина структурується, з тим щоб до них можна було застосувати відомі методи аналізу.

2 ОГЛЯД ПРОЕКТУ DeepQA

Природна мова дуже складна для сприйняття комп’ютером, зважаючи на неявний сенс слів, їх двозначність та велику залежність від контексту. Філософія, що лягла в основу розробки технології DeepQA, полягає в тому, що істинний інтелект виникає лише у результаті взаємодії великої безлічі різних алгоритмів, кожен з яких підходить до даних з різних точок зору.

Жодна програма, що розробляється зверху вниз, не матиме усього необхідного для розуміння природної мови. Адже система повинна розвиватися на основі постійного вкладу різних алгоритмів, які повинні врівноважувати один одного, щоб сформувати цілісну і точну інтерпретацію закладеного значення. Саме архітектура ПЗ для глибокого аналізу змісту і аргументації на основі фактичних даних DeepQA стала втіленням даної філософії [1].

DeepQA - це архітектура із супутньою методологією, що була розроблена компанією IBM як комп’ютерна система, що могла б змагатися на рівні з людиною у режимі реального часу в американській телевізійній вікторині «Jeopardy», проте вона не є специфічною лише для Jeopardy Challenge. IBM розпочали її адаптацію до різних бізнес-додатків та інших дослідницьких завдань, включаючи медицину, корпоративний пошук та ігри.

Основними принципами DeepQA стали масовий паралелізм, наявність великої кількості експертних вузлів, усебічна оцінка достовірності гіпотези, а також інтеграція поверхневих та глибоких знань [1]. Розглянемо їх більш детально:

1. **Масовий паралелізм**.

Використовується при розгляді великої кількості інтерпретацій та гіпотез.

2. **Велика кількість експертних вузлів**.

Полегшують інтеграцію, реалізацію та контекстну оцінку широкого спектру вільно поєднаних імовірнісних питань та контент-аналітики.

3. **Широкомасштабна оцінка достовірності**.

Жоден з компонентів системи не бере на себе відповідальність давати відповідь. Усі компоненти окремо формують характеристики та відповідний

рівень достовірності за допомогою оцінки різних питань та інтерпретацій відповідей на них.

4. **Інтеграція поверхневих та глибоких знань**.

Зберігати баланс між використанням строгої та поверхневої семантики, використовуючи вільно сформовані онтології (ODSE підхід).

2.1 Програмне забезпечення DeepQA

Архітектура DeepQA являє собою потужну систему, що використовує передову обробку природної мови, семантичний аналіз, пошук інформації, автоматизоване мислення і машинне навчання (рис. 2.1.1). DeepQA глибоко аналізує інформацію, що вводиться на природній мові, щоб синтезувати і організовувати відповіді та їх обґрунтування на основі багатьох знань, доступних в комбінації з існуючими текстами природною мовою і базами даних .

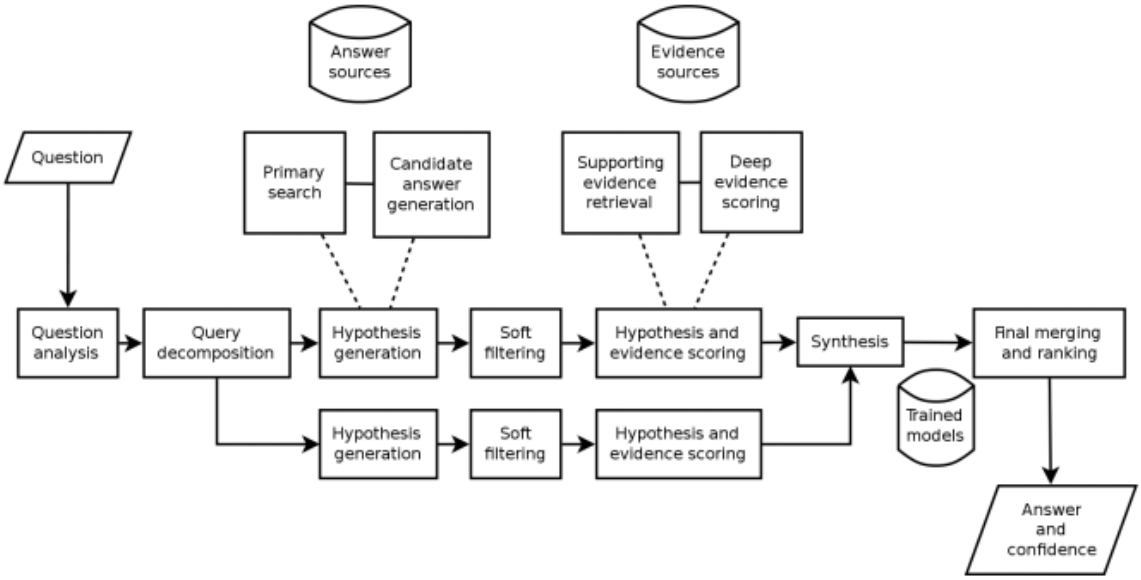

Рисунок 2.1.1 «Високорівнева архітектура IBM DeepQA»

Архітектура DeepQA розглядає проблему автоматичної відповіді на питання як потужні паралельні задачі генерації та оцінки гіпотез. Таким чином, DeepQA – це не просто «питання-відповідь», а система, яка виконує диференційну діагностику:

вона генерує широкий спектр гіпотез, для кожної з яких формує відповідний рівень достовірності, заснований на наявних даних, шляхом збору, аналізу та оцінки матеріалів. Для розуміння процесу роботи даної технології більш детально розглянемо деякі архітектурні ролі [1].

Збір інформації (Content Acquisition)

Після того, як до системи надійшло питання, першим кроком у вирішенні проблеми відповіді на нього є ідентифікація та збір інформації для використання у джерелах інформації (рис. 2.1.1) та формуванні відповідей. Збір інформації – це поєднання багатьох мануальних та автоматичних кроків. Одним з таких є, наприклад, аналіз інших питань даної області для встановлення типу поточного питання, а також характеристики предметної області.

Аналіз питання (Question Analysis)

Першим кроком безпосередньо у ході відповіді на питання (runtime) є аналіз питання, що включає також такі кроки як класифікація питання (Question Classification), виявлення залежностей (Relation Detection) та декомпозиція (Decomposition).

На даному етапі система намагається зрозуміти, яке саме питання було задане та проводить початковий аналіз, під час якого визначається як воно буде оброблене іншими частинами системи. Також даний етап передбачає спільну роботу багатьох експертних вузлів (механізми глибокого та поверхневого семантичного аналізу, розстановки смислових міток, кореляційного аналізу, виявлення зв’язків, іменування об’єктів) системи, за якої у системі впроваджуються поверхневі та глибокі механізми семантичного аналізу, логічні форми, семантичні рольові мітки, зіставлення, іменування сутностей, а також інші специфічні види аналізу для відповіді на питання.

Генерування гіпотези (Hypothesis Generation)

На етапі генерування гіпотези система, використовуючи результати аналізу питань, формує відповіді-кандидати (Candidate Answer Generation), шляхом пошуку джерел в системі та вилучення результатів. Кожна відповідь-кандидат розглядається системою як окрема гіпотеза, що потребує доведення з деяким ступенем достовірності.

Одним з допоміжних процесів на даному етапі роботи системи є первинний пошук (Primary Search). Його мета полягає в тому, щоб знайти якомога більше матеріалів, що потенційно можуть містити відповіді на поставлене питання. Численні механізми глибокого контентного аналізу має вилучити кандидатів та оцінити їх достовірність. Для підвищення точності враховуються усі знайдені докази, які можуть підтвердити або спростувати інформацію, що міститься у відповідях-кандидатах.

М’яка фільтрація (Soft Filtering)

Ключовим кроком у питанні управління ресурсами є застосування менш ресурсоємних алгоритмів ранжування великого набору кандидатів для зменшення їх кількості, перш ніж з ними розпочнуть працювати алгоритми більш точної оцінки. Кандидати, що подолали поріг м'якої фільтрації, переходять до гіпотези та оцінки достовірності, у той час як кандидати, які не подолали цей поріг, переходять безпосередньо на етап злиття (Final Merging). Модель оцінки м'якої фільтрації та поріг фільтрації визначаються на основі тестувань на довільних наборах даних.

Доведення та оцінка гіпотези (Hypothesis and Evidence Scoring)

На даному етапі відповіді-кандидати, які подолали поріг м'якої фільтрації, проходять жорсткий процес оцінки, який включає в себе збір додаткових доказів на підтвердження кожної відповіді-кандидата або гіпотези, а також застосування широкого спектра глибокого аналізу отриманих результатів з метою оцінки доказів. Архітектура підтримує інтеграцію різних методів збору доказів. Однією з особливо ефективних методик є пошук за посиланнями, при якому відповідь-кандидат додається до первинного пошукового запиту в якості необхідного терміну. При цьому будуть отримані уривки, що містять відповідь-кандидата, який використовується у контексті вихідних термінів питання.

Також на даному кроці відбувається оцінка гіпотези – етап, на якому виконується основна частина глибокого аналізу контенту. Алгоритми оцінки визначають ступінь впевненості у тому, що знайдені докази підтверджують відповідь-кандидата. Технологія DeepQA підтримує і заохочує включення безлічі різних компонентів, які враховують різні міри доказів і дають оцінку, яка

відповідає тому, наскільки добре докази підтверджують відповідь-кандидата. DeepQA надає загальний формат для оцінки додаткових доказів , у той же час накладаючи кілька обмежень на оцінку самих гіпотез, – це дозволяє розробникам DeepQA швидко розгортати, поєднувати та налаштовувати компоненти. Наприклад, Watson використовує понад 50 компонентів для підрахунку оцінок, що формуються, ґрунтуючись на даних з різних типів джерел, включаючи неструктурований та напівструктурований текст. Ці оцінювачі враховують такі речі, як ступінь відповідності між структурою кандидата і питанням; надійність джерела гіпотези; геопросторове розташування; тимчасові відносини; таксономічну класифікацію; лексичні і семантичні відносини, у яких бере участь кандидат; кореляцію кандидата з умовами питання; його популярність та інше.

Остаточне злиття та ранжування (Final Merging and Ranking)

Одним з найскладніших завдань для системи є аналіз питання та змісту таким чином, щоб визначити точну відповідь, а також достатній рівень достовірності. Мета етапу остаточного ранжування і злиття – оцінити сотні гіпотез, щоб виявити єдину найбільш імовірну, з урахуванням наявних даних і оцінити її достовірність.

Численні відповіді-кандидати можуть бути еквівалентними, незважаючи на різні форми. Це ускладнює роботу механізмів ранжирування, які використовують відносні відмінності між кандидатами. Без злиття, алгоритми ранжирування будуть порівнювати кілька поверхневих форм, які представляють один і той же відповідь, і намагатися їх розрізняти. Хоча було запропоновано один напрямок досліджень, заснований на підвищенні довіри до схожих кандидатів (Ko, Nyberg, і Luo 2007), підхід DeepQA базується на тому, що різні форми часто призводять до радикально різних, хоча і потенційно взаємодоповнюючих, оцінок. Таким чином, злиття являє собою підхід, який об'єднує бали відповідей до ранжирування і оцінки достовірності. Використовуючи набір алгоритмів зіставлення та нормалізації, Watson ідентифікує еквівалентні та пов'язані з ними гіпотези, а потім дозволяє виконати злиття по кожній функції для об'єднання балів.

Фінальним етапом роботи системи після злиття (Final Merging) є етап ранжування (Ranking) гіпотези та оцінка достовірності на основі їх об'єднаних балів.

Таким чином, архітектура і методологія, розроблені в рамках цього проекту, наголошують на необхідності системного підходу до досліджень в області QA. Фахівці IBM розробили безліч різних алгоритмів для вирішення різного роду проблем в області QA, однак жоден алгоритм їх не вирішує. Комплексні системи, як правило, включають в себе безліч складних та суміжних взаємодій. Дизайн системи та методологія, яка сприяє ефективній інтеграції багатьох імовірнісних компонентів, була вкрай важлива для успіху даного проекту.

3 СУПЕРКОМП’ЮТЕР IBM WATSON

IBM Watson – це суперкомп'ютер компанії IBM розроблений в рамках проекту IBM DeepQA дослідницькою групою на чолі з головним дослідником Девідом Ферруччі. Watson був названий на честь першого виконавчого директора IBM Томаса Дж. Уотсона. IBM Watson був створений, як комп'ютерна система, що відповідає на запитання (QA system), яку компанія IBM створила для застосування сучасних методів обробки природної мови, пошуку інформації, представлення знань, автоматичного обґрунтування та технологій машинного навчання до відповіді на питання. При створенні суперкомп'ютера компанія IBM заявила, що понад 100 різних методів використовуються для аналізу природної мови, виявлення джерел, пошуку та створення гіпотез, пошуку та оцінки доказів, а також об'єднання та ранжування гіпотез. В останні роки можливості IBM Watson були розширені, і методи роботи з ним були змінені, задля того, щоб скористатися перевагами нових моделей розгортання Watson на IBM Cloud, а також розвинути можливості машинного навчання та оптимізували апаратне забезпечення, доступне розробникам та дослідникам. Це вже не просто QA-система, здатна відповідати на поставлені питання. Тепер Watson може бачити, чути, читати, говорити, розуміти, інтерпретувати, вчитися та рекомендувати [2].

3.1 Конструкція та програмна частина IBM Watson

Watson являє собою систему, що складається з трьох компонентів: суперкомп’ютера, що працює під управлінням ОС Linux; ПЗ, що реалізує архітектуру UIMA та системи DeepQA [2]. Центральною частиною і, можливо, найбільш важливою для подальшого розвитку системи є UIMA. Технологія управління неструктурованою інформацією (UIM) і відповідна архітектура UIMA розроблялася в IBM Research ще з 90-х років групою, яка налічувала близько 200 співробітників. Їх діяльність була зосереджена на засобах для роботи з NLP та включала підтримку діалогу на природній мові, виділення корисної інформації, аналіз текстів, класифікацію документів, машинний переклад і QA системи. Підсумком стало створення сполучного ПО, що отримав назву UIMA, яке може служити ядром для створення і впровадження розподілених аналітичних машин

(analysis engine), або UIM-додатків, що дозволяють отримувати знання з неструктурованою інформації, в тому числі з текстів, аудіо, відео та зображень. Структура UIMA складається з декількох компонентів (рис. 2.1.1).

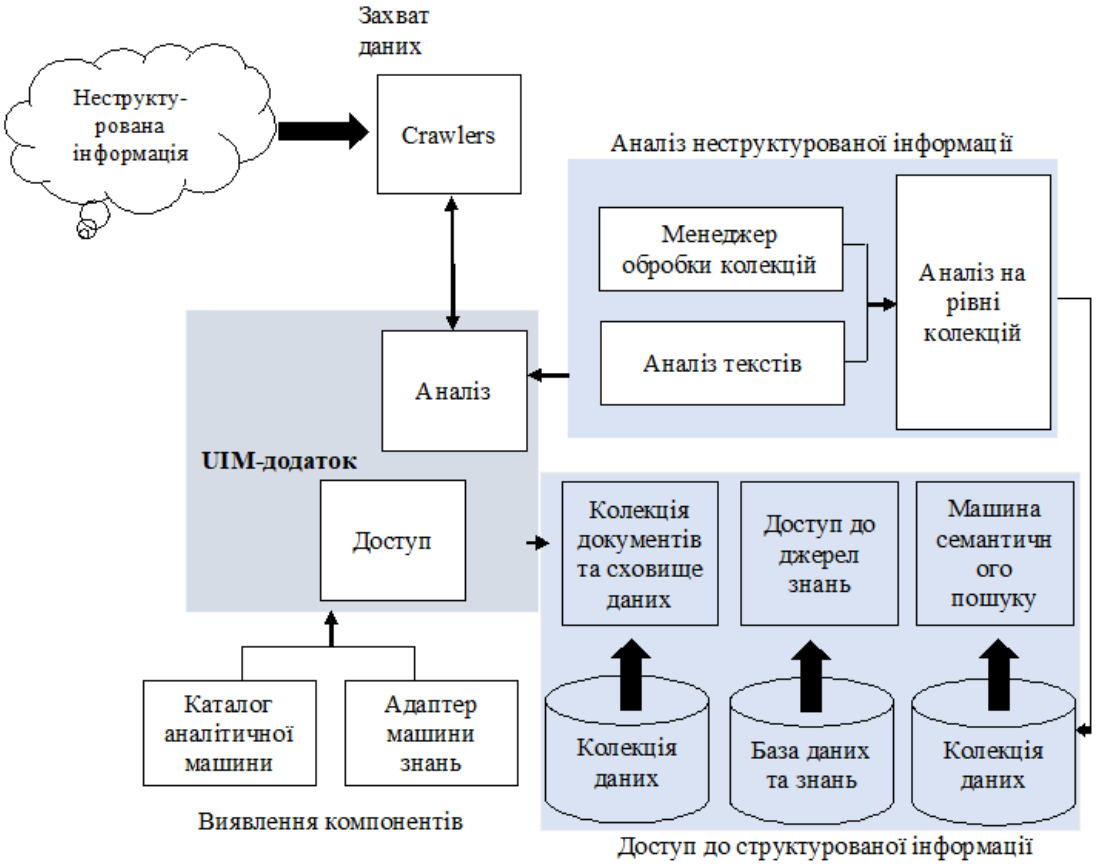

Рисунок 2.1.1 «Архітектура UIMA»

Захоплення даних (Acquisition) забезпечує збір документів з різних джерел і формування необхідних для конкретних програм колекцій. Функцію захоплення можуть, наприклад, здійснювати Web-павуки (web crawler), а також інші засоби, які саме, для додатків не має значення, оскільки є спеціальний рівень інтерфейсу Collection Reader, що зв'язує додатки з колекціями даних і метаданих.

Аналіз неструктурованої інформації (Unstructured Information Analysis) проходить шляхом виконання двох послідовних етапів – аналіз документів, а потім аналіз колекцій документів. Вхідні документи обробляються текстовими аналітичними машинами (Text Analysis Engine), такими як транслятори та модулі граматичного розбору, класифікації, узагальнення. Використовуючи вхідні документи, текстові аналітичні машини виробляють узагальнені аналітичні структури (Common Analysis Structure). На етап аналізу колекцій документи

можуть надходити безпосередньо або через проміжний етап, на якому виконується необхідна фільтрація і переформатування для подальшої паралельної обробки. Аналіз на рівні колекцій (Collection Level Analysis) дозволяє узагальнити відомості, що містяться в колекції документів.

Аналіз структурованої інформації (Structured Information Analysis) використовується як для вхідних даних, що надходять в структурованій формі, так і для даних, що з'являються після аналізу неструктурованої інформації, де їх значна частина структурується, з метою того, щоб до них можна було застосувати відомі методи аналізу. У результаті аналітичні механізми, призначені для двох типів даних, охоплюються загальним зворотнім зв'язком. У 2008 році був випущений реліз Apache UIMA-AS (Asynchronous Scaleout), в якому до основної функціональності UIMA була додана можливість асинхронного масштабування, а також UIMA було узгоджено з Apache Hadoop, що істотно розширило функціональні можливості і сферу застосування паралельних обчислень. Hadoop має ряд принципових відмінностей від традиційних СУБД. Hadoop не зберігає дані, це система обробки даних нового типу. Дане рішення ближче до ОС або сполучного ПЗ, але класичні ОС в кінцевому підсумку обробляють команди і управляють потоками команд, а Hadoop обробляє дані і керує їхніми потоками [2]. Усі вищеперелічені технології використовувались при створенні ПЗ IBM Watson.

З появою версії UIMA-AS відкрилася можливість розпаралелювання задач, тому дія, яка раніше потребувала двох годин для виконання одним процесором, тепер виконується в режимі реального часу. Watson працює на кластері з 10 стійок по 9 стандартних серверів IBM Power 750 на базі процесорів POWER7. Загальна кількість ядер – 2880, і вони розпоряджаються 15 Тбайт оперативної пам'яті, продуктивність обладнання становить 80 TFLOPS (80 трильйонів операцій з плаваючою комою за 1 секунду) [3]. Процесор POWER7 працює на частоті 3,55 ГГц і має 8 ядер, кожне з яких, в свою чергу, апаратно підтримує одночасне виконання чотирьох потоків команд. Систематизована інформація про технічні характеристики суперкомп’ютеру наведена у табл.1. Такий процесор підходить для задач обробки величезних обсягів інформації в паралельному режимі.

Поєднання високої продуктивності ядра Power7 з пам'яттю 512 Гбайт на ядро перетворює апаратну частину Watson в потужний інструмент для підтримки процесів, які потребують великої пам'яті і високої процесорної потужності.

Характеристика	**Значення**
Сервер	Linux-сервер IBM Power 750
К-сть серверів	90
Процесор	IBM POWER7
К-сть ядер	2880
RAM	15 Тб
Продуктивність	80 TFLOPS

Таблиця 1 «Технічні характеристики IBM Watson»

Перевага Watson полягає у тому, що він зібраний з комерційно доступних компонентів, а досвід його створення може бути поширений на інші системи. Кластер працює під управлінням операційної системи SUSE Linux Enterprise Server 11. Комплекс програм, який реалізує численні алгоритми штучного інтелекту (обробка природної мови, вилучення інформації, подання знань, автоматичний логічний висновок і машинне навчання), написаний на мовах Java, C ++ і Prolog.

Одна з найважливіших частин Watson – база знань, підготовка якої була окремою задачею. У якості джерел були обрані енциклопедії, словники, збірки газетних статей і повний текст Вікіпедії. Все це було зібрано в базовий корпус, до якого була застосована процедура автоматичного розширення, в результаті чого з мережі були автоматично обрані тексти, що доповнюють тексти базового корпусу. Система мала доступ до 200 мільйонів сторінок структурованої і неструктурованої інформації загальним об'ємом у 4 Тб [4].

Визнання здатності Watson розуміти сенс і контекст сказаного на природній мові, знаходити точні відповіді на складні питання може змінити уявлення людей про те, для чого можуть бути використані комп'ютери. Watson відмінно

справляється із завданням по розумінню складного питання і пошуку найкращої відповіді. Учені IBM зазначають, що Watson насправді не думає. «Його мета не в тому, щоб моделювати людський мозок» [4], - сказав Девід Ферруччі , який 15 років працював в IBM Research над проблемами природної мови і знаходив відповіді серед неструктурованої інформації. «Мета полягає в тому, щоб створити комп'ютер, який міг би більш ефективно розуміти і взаємодіяти на природній мові, але не обов'язково так, як це роблять люди» [4]. QA системи, у тому числі IBM Watson, не відповідають на питання – вони надають тисячі результатів пошуку, які відповідають ключовим словам.

3.2 Watson Developer Cloud

Спочатку системою Watson оснащувалися лише суперкомпьютери, але з розвитком хмарних платформ, можливості Watson стали широко поширеними та доступними для всіх через Watson Developer Cloud – хмарний сервіс для розробників, що надається за допомогою IBM Bluemix. Bluemix – інновація в серії хмарних рішень IBM. Це середовище дозволяє розробникам і організаціям швидко і легко створювати, розгортати й адмініструвати додатки в хмарі. Bluemix є реалізацією архітектури IBM Open Cloud Architecture на основі відкритого ПЗ Cloud Foundry, що працює за принципом «платформа як послуга» (Platform as a Service - PaaS) [5]. Bluemix надає послуги корпоративного рівня, які можна легко інтегрувати в хмарні додатки, не вдаючись у тонкощі їх встановлення та налаштування.

Watson Developer Cloud надає доступ до набору когнітивних сервісів, які дозволяють розробникам розширювати і створювати для користувача досвід доступу «нового покоління» у додатках, здатних взаємодіяти з людиною. Компанія IBM публікує прикладні програмні інтерфейси (API) у своїй хмарі, які дозволяють користувачам створювати власні додатки ШІ, що використовують основну технологію Watson на серверній частині. Існують API-інтерфейси, які підтримують популярні середовища розробки, такі як Java, Python та інші.

У IBM також є API-конектори для попередньо навчених алгоритмів глибокого навчання, які дозволяють користувачам створювати додатки для таких

завдан, як обробка природної мові, розпізнавання зображень та аналіз тонів [6]. Більшість хмарних сервісів Watson Developer Cloud можуть використовуватися розробниками як когнітивні системи без необхідності в навчанні, деякі з них – його потребують. Навчання у кожному випадку вимагає спеціальних знань в конкретній галузі з відповідними навчальними даними. Без таких знань і навчання в цій області Watson не зможе повністю розкрити свій потенціал за допомогою даних користувача.

Здатність Watson нарощувати людський інтелект залежить від навчання, знань в області і, в кінцевому рахунку, від людського інтелекту. Watson Developer Cloud надає доступ до можливостей Watson без необхідності використання облікового запису IBM Bluemix.

ВИСНОВОК

Штучний інтелект, технології розробки якого дуже стрімко розвиваються і удосконалюються, не можна уявити без розуміння природної мови. Мова служить виразом думки, тому й розумовий процес неможливий без мови, а їх розвиток у процесі еволюції тісно пов'язаний. У даний час дуже складно уявити, що штучно створена система може справлятися з таким завданням, однак, перші кроки в цьому напрямку зроблені і демонструють чудові результати. Одним з таких кроків до створення когнітивних систем стала розробка IBM Watson, що істотно відрізняється від інших рішень.

IBM Watson – суперкомп’ютер, оснащений QA системою штучного інтелекту. Унікальності системі надають такі характеристики як обробка природної мови, формування гіпотез та оцінка їх достовірності, а також здатність динамічного навчання. Після років інтенсивних досліджень та розробок, проведених командою IBM Research, Watson демонструє точність, достовірність та швидкість на рівні експертів, а також може бачити, чути, читати, говорити, розуміти, інтерпретувати, вчитися та рекомендувати.

У ході виконання даної курсової роботи були розглянуті принципи роботи суперкомп’ютеру Watson, що базуються на технології DeepQA, його конструкція та апаратна частина, а також було охарактеризовано процес отримання системою відповіді на поставлене питання. Також було приділено особливу увагу архітектурі технології DeepQA, функція окремих архітектурних ролей, оскільки дана технологія стала базисом для розробки когнітивної системи IBM Watson.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. — СПб.: Питер, 2001. – 384 с
2. Рубашкин, В.Ш. Онтологическая семантика: Знания. Онтологии. Онтологически ориентированные методы информационного анализа текстов / В.Ш. Рубашкин. — М.: ФИЗМАТЛИТ, 2012. – 346 с.
3. Building Watson: An Overview of the DeepQA Project [Електронний ресурс].Режим доступу:https://www.researchgate.net/publication/220605292_Building_Watson_An_Overview_of_the_DeepQA_Project, вільний.
4. It's (not) elementary: How Watson works [Електронний ресурс]. – Режим доступу: https://www.pcworld.com/article/3128401/its-not-elementary-how-watson-works.html, вільний.
5. A Computer Called Watson [Електронний ресурс]. – Режим доступу: https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/ , вільний.
6. IBM Watson supercomputer [Електронний ресурс]. – Режим доступу: https://searchenterpriseai.techtarget.com/definition/IBM-Watson-supercomputer , вільний.
7. Что такое IBM Bluemix? [Електронний ресурс]. – Режим доступу: https://www.ibm.com/developerworks/ru/library/cl-bluemixfoundry/index.html , вільний.
8. Training helps users get more out of Watson Developer Cloud [Електронний ресурс]. – Режим доступу: https://www.ibm.com/blogs/cloud-computing/2017/06/06/training-helps-users-get-watson-developer-cloud/, вільний.

e-mail: ruzudzhenk.jb@gmail.com