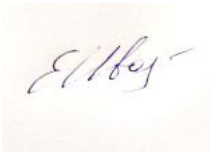
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра системного аналізу та теорії прийняття рішень

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
на тему:
Оцінка емоційного забарвлення текстів на основі засобів і технології аналізу великих даних

студента 4 курсу

Іванченка Георгія
Андрійовича

Науковий керівник:
доктор фізико-математичних наук,
професор

Івохін Є.В.

Робота заслухана на засіданні кафедри системного аналізу та теорії прийняття рішень та рекомендована до захисту в ЕК,

протокол № 10 **від** 07.06. **2022 р.**

Завідувач кафедри системного аналізу та теорії прийняття рішень проф. Наконечний О.Г.


Київ – 2022

Зміст

1. Вступ 3
2. Огляд методів та технологій роботи з великими наборами даних 6
Огляд 6
Архітектура хмарних обчислень 9
Великі дані 15
Методика побудови та зберігання великих даних 20
Аналітичні підходи для обробки великих даних 24
3.Методи та підходи для проведення аналізу емоційного забарвлення текстових даних 29
Методи емоційного аналізу текстових даних 29
Інфраструктурні (кластерні) рішення для аналізу інформації 35
4.Опис реалізованої аналітичної системи 40
5. Висновки 41
Список використаних джерел 42

1. Вступ

Кількість даних, що потребують аналізу зростає кожного дня. Це породжує багато проблем, таких як: обробка та аналіз даних, швидкість аналізу, моніторинг даних. Для вирішення цих проблем були розроблені технології великих даних Big Data - потужний інструмент в руках системного аналітика. Вони допомагають проаналізувати величезні обсяги даних, щоб знайти залежності між елементами системи, знайти сенс в до того протирічних даних, зменшити невизначеність. Аналіз великих даних дозволяє отримати більш повний погляд на прикладні проблеми.

В останні роки було розроблено велику кількість алгоритмів та аналітичних рішень що використовують дані користувача (наприклад вік, стать, статистику використання сервісу тощо). Більшість даних, що оброблюються, мають просту природу - число, дата, належність певній групі ознак. Для таких даних розроблені ефективні засоби аналізу.

Проте є значно складніший тип даних - наприклад текст, відео, аудіо. Ці дані називають неструктурованими і вони представляють особливий інтерес для аналізу, оскільки характеризуються неповнотою, суперечністю та невизначеністю. Саме такі характеристики має задача системного аналізу.

Для розв’язання цієї задачі необхідно розробити систему аналізу, яка здатна в режимі реального часу обробляти велику кількість неструктурованих текстових даних.

Ця задача є особливо цікавою, оскільки саме в такому вигляді люди звикли ділитися своїми думками та враженнями у соціальних мережах, відгуках, блогах тощо. Прикладом задач які вирішуватиме система може бути аналіз ставлення до тих чи інших явищ, подій, ініціатив, відгуків про продукти і т.д. Згодом цю інформацію можна буде використати під час наступних стадій аналізу або прийняття рішень - наприклад морфологічного аналізу. Відгуки про сервіси, закони чи продукти дозволяють визначити вимоги громадян, потреби користувачів, побажання клієнтів; які послуги можна покращити, які нові

послуги або продукти можна створити та якій складовій реформи або закону приділити більшу увагу. Додаткова інформація отримана завдяки аналізу великих даних збільшує прибутки, створює кращі товари та послуги. Саме з цих причин ціль роботи заключається в розробці системи аналізу неструктурованих текстових даних. В ході роботи були вирішені наступні задачі:

- аналіз алгоритмів та підходів до визначення настрою тексту;
-аналіз програмного забезпечення і спеціалізованих сервісів, використовуваних для аналізу великих даних
- розробка веб-порталу для визначення настрою текстових даних
- розробка методу масштабування додатку за допомогою контейнеризації і оркестратора контейнерів;

Результат для практичного застосування - програмна реалізація запропонованого методу, що використовує мови програмування Python, Java, JavaScript:

Мета роботи:

- Розробка ефективної системи аналізу текстових даних для застосування в різних сферах.
- Реалізація веб-порталу для роботи користувача з аналітичною системою.
- Реалізація можливості розгорнути і масштабувати систему на вимогу користувача із застосуванням хмарних технологій

Завдання:

- Дослідження існуючих методів аналізу великих даних.
- Аналіз методів семантичної обробки інформації.
- Аналіз методів збереження і накопичення інформації.
- Інтеграція запропонованої аналітичної системи у веб-портал.

Об’єкт дослідження: веб-портал для аналізу текстових даних як сервіс.

Кваліфікаційна робота складається зі вступу, 4 розділів, висновку, списку використаної літератури. Список використаної літератури включає в себе 37 джерел. Робота виконана на 45 сторінках друкованого тексту.

У першому розділі розглянуто поняття великих даних, історію його виникнення, проаналізовано основні їх функції. Описано приклади застосування технології великих даних і їх актуальність. Описано архітектуру хмарних обчислень, наведено класифікацію хмарних інфтраструктур за моделлю обслуговування, моделлю розгортання. Детально досліджено найважливіші характеристики хмарних обчислень. Описані головні діючі суб'єкти, актори і споживачі хмарних технологій, їх головні ролі та функції. Детально описано поняття великих даних, історію його виникнення. Досліджено засоби роботи з великими даними. Наведено огляд методик побудови та зберігання великих даних. Описано формати зберігання даних, їх специфіку і особливості, переваги та недоліки застосування тих чи інших форматів. Досліджено аналітичні підходи для обробки великих даних. Дослідження алгоритмів аналізу даних, найкращих напрямків застосування цих алгоритмів. Оглянуто особливості аналізу даних в межах хмарних середовищ.

У третьому розділі описано методи емоційного аналізу текстових даних. Наведено основні типи задач, класифікацію емоцій за Екманом. Досліджено підходи до вилучення емоцій з текст. Описано інфраструктурне рішення для аналізу інформації, розроблене при виконанні кваліфікаційної роботи.

У четвертому розділі наведено опис реалізованої аналітичної системи.

2. Огляд методів та технологій роботи з великими наборами даних

Огляд

Концепція Big Data існує протягом багатьох років; більшість організацій сьогодні розуміють, що якщо вони зберігають всі дані, що отримують впродовж життя свого бізнесу, вони можуть застосувати аналітику та отримати з неї значну користь. Але навіть у 1950-х роках, за десятиліттями, ще до того коли хтось перший використав термін "великі дані", бізнес використовував базову аналітику (по суті числа в електронній таблиці, які аналітик перевіряв вручну), щоб розкрити закономірності, взаємозв'язки та тенденції. Однак переваги, які аналітика великих даних приносить в аналіз - це швидкість та ефективність. Тоді як кілька років тому бізнес міг би зібрати інформацію, залучити аналітика та розкрити тенденції, які можна використати для майбутніх рішень, сьогодні бізнес може отримати поради щодо негайних дій. Здатність працювати швидше - і залишатися гнучкими - надає організаціям конкурентну перевагу, яку вони не мали раніше. Швидкість обробки особливо важлива у зв'язку зі зростаючою тенденцією по накопиченню даних.

Згідно з оцінками, до 2009 року майже всі сектори економіки США мали в середньому 200 терабайт збережених даних (обсяг удвічі більший за складську базу даних найбільшої мережі роздрібних магазинів в США - Wal-Mart - станом на 1999 рік) для компаній з більш ніж 1000 працівниками. За іншими оцінками очікується значне зростання потреби в аналітиках та програмних комплексах що здатні працювати з великими даними. Так відсоток даних, які будуть корисними для аналізу зросте з 22% до більш ніж 35%. Очікується, що всесвітній ринок великих даних (технологій та послуг з їх обробки) зростатиме зі швидкістю близько 23% щорічно в період між 2014 і 2019 роками, а світові доходи від великих даних та бізнес-аналізу збільшиться більш ніж на 50% з майже 122 мільярдів доларів США у 2015 році до більш ніж 187 мільярдів доларів США в 2019 році. Найбільші сектори в яких використовується BD включають

виробництво, банківська справа та страхування, телекомунікації, охорону здоров'я, транспорт та роздрібну торгівлю. Аналітика великих даних допомагає організаціям використовувати впродовж своєї роботи дані для виявлення нових можливостей. Це, в свою чергу, призводить до прийняття більш інформованих бізнес рішень, проведення ефективніших операцій, отримання вищого прибутку та щасливих клієнтів.

У своєму дослідженні "Великі Дані у великих компаніях"[1], директор з досліджень IIA Том Давенпорт розглянув більше 50 підприємств, щоб зрозуміти, як вони використовували великі дані. Він виявив, що BD приносять користь такими способами:

Зниження вартості. Hadoop та хмарна аналітика, приносять значні переваги при зберіганні великої кількості даних, а також можуть виявити більш ефективні способи ведення бізнесу;

Швидше, краще прийняття рішень - завдяки швидкості роботи Hadoop та аналізу пам'яті в поєднанні з можливістю аналізу нових джерел даних підприємства можуть негайно аналізувати інформацію та приймати рішення на підставі того, що вони дізналися;

Нові продукти та послуги - завдяки здатності оцінювати потреби клієнтів та їх задоволення від продукту з’являється можливість надати клієнтам те, що вони хочуть.

Окрім вигоди для бізнесу, аналіз великих даних може бути застосований і в інших областях. За його допомогою можна оцінювати якість освіти, глобальну зміну клімату, глобальні тенденції щодо зайнятості, екологічну ситуацію тощо.

Як і багато нових інформаційних технологій, великі дані можуть призвести до значного скорочення часу, необхідного для обчислень, або створення нових продуктів та послуг. Як і традиційна аналітика, BD здатна підтримувати внутрішні бізнес-рішення. Технології та концепції на яких будується аналіз великих даних організації дозволяють досягати поставлених цілей. Зниження вартості за допомогою BD Організації, орієнтовані на скорочення витрат, прийняли рішення прийняти інструменти великих даних для роботи з даними в межах своїх підрозділів інформаційних технологій керуючись техніко-

економічним критерієм. Зниження вартості може бути додатковою метою після досягнення інших цілей. Скажімо, перша мета організації - інновації в продуктах та послугах, що були отримані за допомогою BD. Після досягнення цієї мети вона може захотіти зменшити витрати.

Другою метою BD є скорочення часу. Мережа універмагів Масу`s змогла скоротити час на ціноутворення для 73 мільйонів одиниць товару з 27 годин до 1 години. Ця можливість дозволяє їй проводити переоцінку товарів набагато частіше й адаптуватися до зміни умов на роздрібному ринку. Розроблена система аналізу бере дані з Hadoop кластера та поміщає його в програмне забезпечення що дозволяє розпаралелити обчислення. Масу`s стверджує що завдяки цьому витрати на апаратне забезпечення були скорочені на 70%.

Ще однією ключовою метою є можливість взаємодії з клієнтом у режимі реального часу, використовуючи аналітику та дані, отримані від досвіду клієнтів. Аналіз зворотнього зв’язку допомагає виправити недоліки власного продукту та уникнути помилок, зроблених конкурентами. Створення нових пропозицій Одна з найамбіційніших задач, яку можна розв’язати за допомогою великих даних - використати їх розробки нових продуктів та послуг. Багато компаній, які використовують цей підхід - онлайн-фірми, що отримують прибуток від продуктів та послуг пов’язаних з даними. Яскравим прикладом може бути LinkedIn, який використовував BD для розробки широкого спектру пропозиції продуктів і нових сервісів, у тому числі люди, яких ви можете знати, групи, які можуть вам сподобатися, хто переглядав мій профіль та інші. Ці сервіси допомогли завоювати мільйони нових клієнтів. Основна мета традиційної аналітики так званих "малих даних" - підтримка внутрішніх бізнес-рішень Які пропозиції цікавлять клієнта? Хто перестане користуватися сервісом найближчим часом? Який об’єм товару необхідно утримувати на складі? Яку ціну можна встановити на свій товар? Ці типи рішень використовують великі дані, якщо є нові, неструктуровані джерела даних, що можуть допомогти знайти рішення. Наприклад, будь-які дані, які можуть допомогти дізнатися чи задоволені клієнти допомагають у прийнятті рішень. Більшість з них - неструктуровані текстові дані.

Архітектура хмарних обчислень

Хмарні обчислення (англ. Cloud computing) - це модель забезпечення повсюдного доступу до мережі на вимогу до загального пулу (англ. Pool) сконфігурованих обчислювальних ресурсів, наприклад, мереж передачі даних, серверів, пристроїв зберігання даних, додатків і сервісів - як разом, так і окремо, які можуть бути оперативно надані та звільнені з мінімальними експлуатаційними витратами і / або зверненнями до провайдера. При цьому у користувача - клієнта фактично залишається лише інтерфейс його інформаційної системи, а його дані, які він використовував, програмні засоби, інформаційна інфраструктура перебувають у провайдера.

До основних причин виникнення та просування "хмарних" технологій можна віднести такі:

- природний надлишок обчислювальних потужностей і пам'яті суперкомп'ютерів, підключених до мережі каналами високої пропускної здатності (пошукові системи, потужні хостинги і ін.);
- бажання власників цих потужностей отримати від них прибуток;
- Споживачі хмарних обчислень можуть значно зменшити витрати на інфраструктуру інформаційних технологій (в короткостроковому - для "стартапів", і середньостроковому планах) і гнучко реагувати на зміни обчислювальних потреб, використовуючи властивості обчислювальної еластичності (англ. Elastic computing) хмарних послуг, але повністю втрачають свою незалежність : інформаційну, стратегічно ділову, ідеологічну, політичну.

Хмарні технології з'явилися зовсім недавно: у 2006 році один з найбільших американських інтернет-магазинів Amazon надав свої обчислювальні ресурси, що не використовуються (а на той час їх обсяг став величезним) абсолютно новим чином. Традиційно для оренди ресурсів у дата-центрах необхідно було скласти договір та внести плату за певний термін. Лінійка типорозмірів серверів (обсяг оперативної пам'яті, кількість ядер, розмір дискового простору та ін)

досить велика і вибирається заздалегідь, до підписання договору. Можна орендувати багато серверів, пов'язати їх високопродуктивною мережею, підключити балансувальник навантаження та отримати систему, що обробляє велике навантаження. У подібній моделі використання ресурсів є суттєві незручності. При створенні програм часто невідомо, яка буде потрібно навантаження, на який термін орендувати сервери програми. Або такий приклад: створюється стартап, орендуються сервери і до закінчення терміну оренди цей стартап "вмирає". Що робити з непотрібними орендованими серверами? Ще складніше справа з покупкою фізичних серверів. Адже їх, крім адміністрування операційної системи та встановлених додатків, необхідно обслуговувати фізично. Сюди входить підбір приміщення, електроживлення, системи охолодження, вентиляції. Всі ці проблеми можна вирішити за допомогою еластичних обчислювальних ресурсів, що надаються хмарними провайдерами. (Платформа Amazon Web Services називає ці ресурси EC2 - Elastic Cloud Computers.) Ключові переваги хмарної моделі такі: ресурси надаються на вимогу і таким чином звільняються; плата нараховується за фактичний час використання ресурсів; надання та звільнення ресурсів провадиться самим споживачем ресурсів через веб-портал, без будь-якої паперової тяганини з договорами.

Моделі обслуговування

Software-as-a-Service

Програмне забезпечення як послуга (SaaS, англ. Software-as-a-Service) - модель, в якій споживачеві надається можливість використання прикладного програмного забезпечення провайдера, який працює в хмарній інфраструктурі і доступного з різних клієнтських пристроїв або за допомогою тонкого клієнта, наприклад, з браузера (наприклад, веб-пошта) або інтерфейс програми. Контроль і управління основною фізичною і віртуальною інфраструктурою хмари, в тому числі мережі, серверів, операційних систем, зберігання, або навіть індивідуальних можливостей додатка (за винятком обмеженого набору

призначених для користувача налаштувань конфігурації програми) здійснюється хмарним провайдером.

Platform-as-a-Service

Платформа як послуга (PaaS, англ. Platform-as-a-Service) - модель, коли споживачеві надається можливість використання хмарної інфраструктури для розміщення базового програмного забезпечення для подальшого розміщення на ньому нових або існуючих додатків (власних, розроблених на замовлення або придбаних тиражованих додатків). До складу таких платформ входять інструментальні засоби створення, тестування і виконання прикладного програмного забезпечення - системи управління базами даних, сполучне програмне забезпечення, середовища виконання мов програмування - надаються хмарним провайдером. Контроль і управління основною фізичної і віртуальної інфраструктурою хмари, в тому числі мережі, серверів, операційних систем, зберігання здійснюється хмарним провайдером, за винятком розроблених або встановлених додатків, а також, по можливості, параметрів конфігурації середовища (платформи).

Infrastructure-as-a-Service

Інфраструктура як послуга (IaaS, англ. IaaS or Infrastructure-as-a-Service) надається як можливість використання хмарної інфраструктури для самостійного управління ресурсами обробки, зберігання, мереж і іншими фундаментальними обчислювальними ресурсами, наприклад, споживач може встановлювати і запускати довільне програмне забезпечення, яке може включати в себе операційні системи, платформенне і прикладне програмне забезпечення. Споживач може контролювати операційні системи, віртуальні системи зберігання даних і встановлені програми, а також обмежений контроль набору доступних сервісів (наприклад, міжмережевий екран, DNS). Контроль і управління основними фізичною і віртуальною інфраструктурою хмари, в тому числі мережі, серверів, типів використовуваних операційних систем, систем зберігання здійснюється хмарним провайдером.

Моделі розгортання

Private cloud

Приватна хмара, - інфраструктура, призначена для використання однією організацією, що включає декілька споживачів (наприклад, підрозділів однієї організації), можливо також клієнтів і підрядників даної організації. Приватна хмара може перебувати у власності, управлінні та експлуатації як самої організації, так і третьої сторони (або будь-якої їх комбінації), і вона може фізично існувати як всередині, так і поза юрисдикцією власника.

Public cloud

Публічна хмара - інфраструктура, призначена для вільного використання широкою публікою. Публічна хмара може перебувати у власності, управлінні та експлуатації комерційних, наукових і урядових організацій (або будь-якої їх комбінації). Публічна хмара фізично існує в юрисдикції власника - постачальника послуг.

Hybrid cloud

Гібридна хмара - це комбінація з двох або більше різних хмарних інфраструктур (приватних, публічних або суспільних), що залишаються унікальними об'єктами, але пов'язаних між собою стандартизованими або приватними технологіями передачі даних і додатків (наприклад, короткочасне використання ресурсів публічних хмар для балансування навантаження між хмарами).

Community cloud

Хмара спільноти - вид інфраструктури, призначена для використання конкретним співтовариством (кланом) споживачів з організацій, що мають спільні завдання (наприклад, місії, вимоги безпеки, політики, і відповідності різним вимогам). Така хмара може перебувати в кооперативній (спільної)

власності, управлінні та експлуатації однієї або більше з організацій спільноти або третьої сторони (або будь-якої їх комбінації), і вона може фізично існувати як всередині, так і поза юрисдикцією власника.

Характеристики хмарних обчислень

Національним інститутом стандартів і технологій США зафіксовані такі обов'язкові характеристики хмарних обчислень:

- Самообслуговування на вимогу (англ. Self service on demand), споживач самостійно визначає і змінює обчислювальні потреби, такі як серверний час, швидкості доступу та обробки даних, обсяг збережених даних без взаємодії з представником постачальника послуг;
- Універсальний доступ по мережі, послуги доступні споживачам через мережу передачі даних незалежно від використовуваного термінального пристрою;
- Об'єднання ресурсів (англ. Resource pooling), постачальник послуг об'єднує ресурси для обслуговування великого числа споживачів в єдиний пул для динамічного перерозподілу потужностей між споживачами в умовах постійної зміни попиту на потужності; при цьому споживачі контролюють тільки основні параметри послуги (наприклад, обсяг даних, швидкість доступу), але фактичний розподіл ресурсів, що надаються споживачеві, здійснює постачальник (в деяких випадках споживачі все-таки можуть управляти деякими фізичними параметрами перерозподілу, наприклад, вказувати бажаний центр обробки даних з міркувань географічної близькості);
- Еластичність, послуги можуть бути надані, розширені, звужені в будь-який момент часу, без додаткових витрат на взаємодію з постачальником, як правило, в автоматичному режимі;
- Облік споживання, постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції (наприклад, обсяг збережених даних, пропускна здатність, кількість користувачів, кількість транзакцій), і на

основі цих даних оцінює обсяг наданих споживачам послуг.

- З точки зору постачальника, завдяки об'єднанню ресурсів і непостійного характеру споживання з боку споживачів, хмарні обчислення дозволяють економити на масштабах, використовуючи менші апаратні ресурси, ніж були потрібні б при виділених апаратних потужностях для кожного споживача, а за рахунок автоматизації процедур модифікації виділення ресурсів істотно знижуються витрати на абонентське обслуговування.
- З точки зору споживача, ці характеристики дозволяють отримати послуги з високим рівнем доступності (англ. High availability) і низькими ризиками непрацездатності, забезпечити швидке масштабування обчислювальної системи завдяки еластичності без необхідності створення, обслуговування і модернізації власної апаратної інфраструктури.
- Зручність і універсальність доступу забезпечується широкою доступністю послуг і підтримкою різного класу термінальних пристроїв (персональних комп'ютерів, мобільних телефонів, інтернет-планшетів).

Головні діючі суб'єкти

- Хмарний Споживач (Cloud Consumer) Особа або організація, що підтримує бізнес-відносини і використовує послуги Хмарних Провайдерів.
- Хмарний Провайдер (Cloud Provider) Особа, організація або сутність, що відповідає за доступність хмарної послуги для Хмарних Споживачів.
- Хмарний Аудитор (Cloud Auditor) Учасник, який може виконувати незалежну оцінку (assessment) хмарних послуг, обслуговування інформаційних систем, продуктивності і безпеки реалізації хмари.
- Хмарний Брокер (Cloud Broker) Сутність, керуюча використанням, продуктивністю і наданням хмарних послуг, встановлює відносини між хмарними Провайдерами і хмарними Споживачами.
- Хмарний Оператор Зв'язку (Cloud Carrier) Посередник, який надає послуги підключення та транспорту (послуги зв'язку) <доставки> хмарних послуг

від Хмарних Провайдерів до Хмарних Споживачів.

Актори, їх ролі та функції

- Хмарний Споживач - Особа або організація, що підтримує бізнес-відносини і використовує послуги Хмарних Провайдерів.
- Хмарні споживачі категоризуються за трьома групами, заснованим на їх додатках / різних сценаріях використання.

Основна діяльність (активності) користувачів

- SaaS - Використовує додатки / сервіси для автоматизації бізнес-процесів (Бізнес-користувачі, адміністратори додатків)
- PaaS - Розробляє, тестує, розгортає і управляє програмами, розгорнутими в хмарному оточенні
  (Розробники додатків, тестувальники, адмінісратори)
- IaaS - Створює / встановлює, управляє і моніторить сервіси для управління IT-інфраструктурою
  (Системні розробники, адміністратори, IT-менеджери)

Великі дані

Поняття Великих даних

Концепція Великих даних не нова, вона виникла за часів мейнфреймів і пов’язаних з ними наукових обчислень [2, 3]. Як добре відомо, наукомісткість обчислень завжди було складним завданням. Як правило, вона нерозривно пов’язана з обробкою великих обсягів інформації. Проте, безпосередньо термін «Великі дані» (Big Data) з’явився порівняно недавно. Він є одним з небагатьох, що має відомий день народження – 3 вересня 2008 р. Тоді було випущено

спеціальний випуск найстарішого британського наукового журналу Nature. Журнал присвячений пошукам відповіді на питання: «Як технології можуть вплинути на наукове майбутнє, що відкриває можливості для роботи з Великими даними».

Згідно зі звітом McKinsey інституту під назвою «Великі дані: Наступний рубіж для інновацій, конкуренції і продуктивності», термін «Великі дані» відноситься до наборів даних, розмір яких перевищує ємність звичайної бази даних (БД) для видобування, зберігання, управління і аналізу інформації .

Глобальні сховища даних продовжують зростати. Представлений звіт аналітичної компанії IDC під назвою «Digital Universe Study» в середині 2011 року (який був організований компанією EMC) передбачає, що загальний світовий обсяг генерованих і тиражованих даних може досягати в 2011 році близько 1,8 зеттабайт (1,8 трлн гігабайт). Це приблизно в 9 разів більше, ніж те, що було створено в 2006 році. Проте, концепт «Великі дані» означає набагато більше, ніж просто величезні обсяги інформації. Проблема полягає не в тому, що організації генерують величезні обсяги даних, але більшість з них представлені в форматі, який не дуже добре вписується в традиційний структурований формат бази даних. Це веб-журнали, відео, текстові документи, машинний код, або, наприклад, картографічні дані. У результаті, корпорація може мати доступ до величезного обсягу своїх даних і не мати необхідних інструментів для встановлення зв’язків між цими даними, автоматизованого формулювання конструктивних висновків на основі аналізу цих даних. Крім того, дані зараз оновлюються частіше, і ми маємо ситуацію коли традиційні методи аналізу інформації не можуть опрацювати величезні обсяги даних, що постійно оновлюються, і це, в кінцевому рахунку, прокладає шлях для технологій Big Data.

EWeek подає визначення, запропоноване дослідницькою компанією Gartner: «Великі дані характеризуються обсягом, різноманітністю і швидкою плинністю структурованих і неструктурованих даних в процесорах і пристроях зберігання даних, а також перетворення даних для задач бізнес-консалтингу для підприємств»[4].

Великі дані (Big Data) в інформаційних технологіях за визначенням К. Лінч, Д. Леней – набір методів та засобів опрацювання структурованих і неструктурованих різнотипних динамічних даних великих обсягів з метою їх аналізу та використання для підтримки прийняття рішень [3]. Є альтернативою традиційним системам управління базами даних і рішенням класу Business Intelligence. До цього класу відносять засоби паралельного опрацювання даних (NoSQL, алгоритми MapReduce, Hadoop) [5, 6, 7, 8]. На думку компанії DCA (Data-Centric Alliance) під Big Data розуміють не якийсь конкретний об'єм даних і навіть не дані, а методи їх обробки, які дозволяють розподілено обробляти інформацію [9]. Ці методи можна застосовувати як до великих масивів даних (таких як дані всіх сторінок в мережі Інтерет), так і до малих масивів (інформація про денні поступлення товару в магазин).

Визначальними характеристиками для Великих даних є обсяг (volume, в сенсі величини фізичного обсягу), швидкість (velocity в сенсах як швидкості приросту, так і необхідності високошвидкісної обробки таотримання результатів), різноманіття (variety, в сенсі можливості одночасної обробки різних типів структурованих і слабоструктурованих даних).

Засоби роботи з Великими даними

Засоби для Великих даних	**Опис**
Засоби аналізу даних	
Ambari http://ambari.apache.org	Інструмент веб для надання послуг, управління та моніторингу Apache Hadoop кластерів
Avro http://avro.apache.org	Система серілізації даних
Chukwa http://incubator.apache.org/chukwa	Система колекціонування даних для керування великими розподіленими системами
Hive http://hive.apache.org/	Інфраструктура сховища даних,

	яка забезпечує агрегацію даних
Pig http://pig.apache.org	Високорівнева мова потоків даних і виконуваний framework для паралельних обчислень
Spark http://spark.incubator.apache.org	Швидкий і генеральний обчислювач для даних Hadoop. Забезпечує просту і виразну модель програмування, яка підтримує широкий спектр додатків, у тому числі ETL, машинного навчання, опрацювання потоків
Spark http://spark.incubator.apache.org	Швидкий і генеральний обчислювач для даних Hadoop. Забезпечує просту і виразну модель програмування, яка підтримує широкий спектр додатків, у тому числі ETL, машинного навчання, опрацювання потоків
ZooKeeper http://zookeeper.apache.org/	Високопродуктивна служба координації для розподілених застосунків
Actian http://www.actian.com/about-us/#overview	Забезепечує зберігання сирих даних і готує дані для подальшого аналізу
HPCC http://hpccsystems.com	Забезпечує швидке перетвореня, паралельне опрацювання для застосувань з Великими даними

Засоби Data Mining	
Orange http://orange.biolab.si	Візуалізація та аналіз даних для новачка і експертів

Mahout http://mahout.apache.org	Бібліотека засобів машинного навчання та видобування даних
KEEL http://keel.es	Еволюційний алгоритм для проблем видобування даних

Засоби Data Mining	
Orange http://orange.biolab.si	Візуалізація та аналіз даних для новачка і експертів
Mahout http://mahout.apache.org	Бібліотека засобів машинного навчання та видобування даних
KEEL http://keel.es	Еволюційний алгоритм для проблем видобування даних

Засоби соціальних мереж	
Apache Kafka	Платформа з високою пропускною здатністюдля опрацювання даних в режимі реального часу
Засоби BI	
Talend http://www.talend.com	Інтеграція даних, управління, інтеграція застосувань, засоби і сервіси для Великих даних
Jedox http://www.jedox.com/en	Функції аналізу, звітності, планування
Pentaho http://www.pentaho.com	Інтеграція даних, бізнес-аналіз, візуалізація даних, прогнозування
Rasdaman http://rasdaman.eecs.jacobs-university.de/	Багатовимірні расторові дані (масив) безобмежень на розмір, наявність мови запитів
Засоби пошуку	
Apache Lucene http://lucene.apache.org	Застосування для повнотекстового індексування і пошуку

Apache Solr http://lucene.apache.org/solr	Повнотекстовий пошук, фасетний пошук, динамічна кластеризація, формати документів типу Word, PDF, просторовий пошук
Elasticsearch http://www.elasticsearch.org	Засіб розподіленого повнотекстового пошуку з веб-інтерфейсом і JSON докуметами
MarkLogic http://developer.marklogic.com	NoSQL і XML база даних
mongoDB http://www.mongodb.org	Крос-платформенна документо-орієнтована система управління базами даних з підтримкою JSON та динамічних схем
Cassandra http://cassandra.apache.org	Маштабована мульти-майстерна база даних без єдиної точки відмови
HBase http://hbase.apache.org	Маштабована розподілена база даних з підтримкою структурованого зберігання даних великого обсягу
InfiniteGraph http://www.objectivity.com	Розподілена графова база даних

Методика побудови та зберігання великих даних

Формати зберігання даних

Табличний формат. Найбільш типовий випадок використання текстових файлів зберігання великих даних — зберігання логів додатків у тому чи іншому текстовому форматі. Такі файли можуть мати розширення .log, .txt, .csv та ін. Спільне у цих форматів те, що логи в них, по суті, зберігаються у вигляді таблиці (звідси і назва – табличний формат), кожен рядок якої – запис конкретної події. Рядок складається з набору стовпців, розділених символами.

Це можуть бути прогалини, символи табуляції, двокрапка, крапка з комою і т.д. Подібні файли можна відкрити за допомогою будь-якого текстового редактора (якщо вони не дуже великі), і для їх аналізу (наприклад, пошуку запитів з 500 помилками) підійдуть стандартні утиліти командної оболонки (grep, awk тощо) або ж спеціалізовані послуги аналітики.

Крім того, таблична структура такого файлу дуже зручна для імпорту в реляційну або нереляційну СУБД табличного типу. Особливість таких файлів - лінійна структура зі строго однаковою кількістю стовпців у кожній записи та у загальному випадку лінійний час доступу до записів. Строго кажучи, можливі ситуації, коли різні рядки можуть містити різну кількість стовпців. Наприклад, запис у лог-файлі, що відображає помилку програми та трасування стека, може включати набагато менше рядків, ніж запис, що відображає будь-яка подія, що характеризує нормальну роботу програми. У загальному випадку записи логів містять як мінімум тимчасову мітку. Інші поля (унікальний ідентифікатор, URL, повідомлення помилки тощо) можуть бути відсутніми. Виникає питання: чи корисний запис з одним полем (тимчасовий міткою)? Так. Скажімо, необхідно проаналізувати відвідуваність сайту часу. Підсумувати запити для заданого часового інтервалу (наприклад, 15 хвилин) допоможе саме тимчасова позначка. Табличні файли мають і низку незручностей у використанні. По-перше, для доступу до елемента інформації необхідно знати номер рядка та номер стовпця. Немає універсального стандарту або мови запитів (за винятком SQL-подібної мови спеціалізованих сервісів аналітики), що дуже ускладнює аналіз логів. По-друге, у таких форматах дуже просто додати новий запис в кінець файлу, але дуже важко і витратно вставити, видалити або змінити довільний рядок. Як правило, всі програмні бібліотеки вводу-виводу дозволяють додавати рядок у кінець файлу за допомогою стандартних засобів, але для довільної маніпуляції даними програмісту необхідно буде створити окрему логіку в коді, і це логіка дуже непроста.

Структурований формат. Крім текстового файлу з табличною структурою, поширене зберігання інформації у структурованих форматах JSON, XML. Їхні переваги в тому, що вони дуже зручні для серіалізації/десеріалізації інформації в об'єктно-орієнтованому вигляді в коді програми. Крім того, для обох форматів існують стандарти виконання запитів на вибірку даних (для XML - XPath, XQuery, для JSON - JSONPath, JSONQuery).

JSON є форматом, при якому дані зберігаються в текстовому вигляді як об'єкт

JavaScript. Перевага цього формату текстового файлу – можливість опису структур довільної глибини вкладеності (об'єкт всередині об'єкта, колекція всередині об'єкта та ін.), що неможливо у разі табличного уявлення.
Крім того, суттєво спрощується серіалізація та десеріалізація об'єктів, особливо з JavaScript-програми. Обробка JSON-файлів у спеціалізованих СУБД (DocumentDB) описана в книзі нижче, а деякі реляційні бази даних (наприклад, PostgresSQL) широко підтримують цей формат.

Наступний популярний формат зберігання даних у текстових файлах – XML (eXtensible Markup Language - мова розмітки, що розширюється). Традиційно він використовувався для розмітки (Markup), тобто структурування текстових документів. Термін «мова» (language) говорить про те, що XML містить строгий набір синтаксичних правил, на підставі яких можна побудувати конкретне розширення (extension) цієї мови. Отже, мова XML складається із загальних правил побудови синтаксису, а розширення цієї мови є конкретним набором вкладених тегів та відповідних їм атрибутів, що забезпечує впорядковане представлення конкретної структурованої інформації

Методи зберігання великих даних

На даний момент існують два добре встановлених методи зберігання великих даних:
Складське зберігання (Warehouse storage). Подібно до складу для зберігання фізичних товарів, сховище даних є великим будівельним об’єктом, основною функцією якого є зберігання та обробка даних на рівні підприємства. Це важливий інструмент для аналізу великих даних. Ці великі сховища даних підтримують різні звіти, бізнес-аналітику (BI), аналітику, аналіз даних, дослідження, кібермоніторинг та інші пов’язані види діяльності. Ці сховища, як правило, оптимізовані для збереження та обробки великих обсягів даних у будь-який час, надаючи їх і виводячи через онлайн-сервери, де користувачі можуть отримати доступ до своїх даних без затримок.

Інструменти сховища даних дають змогу ефективніше керувати даними, оскільки дають змогу знаходити, отримувати доступ, візуалізувати та аналізувати дані, щоб приймати кращі бізнес-рішення та досягати більш бажаних бізнес-результатів. Крім того, вони створені з урахуванням експоненційного зростання даних. Немає ризику, що склади будуть захаращені збільшенням обсягу даних, які зберігаються.

Найбільшою перевагою сховищ даних є можливість перетворення необроблених даних в інформацію та розуміння. Сховища даних пропонують ефективний спосіб підтримки запитів, аналітики, звітності, а також надання прогнозів і тенденцій на основі зібраних даних. Дизайн і очищення даних повинні підтримуватися правильним сховищем. Зазвичай сховища даних залежать від великих місткостей, які є надійними, мають нижчі витрати та добре працюють.

Можливо, ви чули про термін « Hadoop », який час від часу зустрічається, але все ще не знаєте, що це таке, і це нормально. Хоча це ціла тема сама по собі, ми пояснимо її коротко. Hadoop — це програмна платформа, призначена для розподіленого зберігання та обробки великих даних для обробки величезних обсягів даних і обчислень. Hadoop робить революцію в аналітиці великих даних для корпоративного сховища.

Хмарне зберігання (Cloud storage). Іншим методом зберігання величезних обсягів даних є хмарне сховище, яке знайоме більшій кількості людей. Якщо ви коли-небудь використовували iCloud або Google Drive, це означає, що ви використовували хмарне сховище для зберігання документів і файлів. Завдяки хмарному сховищу дані та інформація зберігаються в електронному вигляді в Інтернеті, де до них можна отримати доступ з будь-якого місця, що не вимагає прямого доступу до жорсткого диска або комп’ютера. Завдяки такому підходу ви можете зберігати в Інтернеті практично безмежну кількість даних і отримувати до них доступ.

Хмара забезпечує не тільки легкодоступну інфраструктуру, але й можливість швидко масштабувати цю інфраструктуру, щоб керувати великим збільшенням трафіку або використання.

Хмара також забезпечує легкий доступ і зручність використання. Якщо ви хочете отримати доступ до своїх даних у хмарі, все, що вам потрібно зробити, це ввести свої облікові дані, і ви отримаєте доступ. Все, що вам потрібно, це підключення до Інтернету та пристрій для доступу до хмари, наприклад мобільний телефон або портативний комп'ютер. Хмарне сховище значно підвищило продуктивність і ефективність бізнесу, оскільки співробітники можуть миттєво ділитися файлами, отримувати доступ та редагувати їх віддалено.

На додаток до попередніх переваг, хмарне сховище також значно дешевше, ніж фізичне зберігання даних. Сховища даних споживають велику кількість енергії, простору, ресурсів і супроводжуються більшим ризиком. Однак із хмарним сховищем можна заощадити значну суму.

Аналітичні підходи для обробки великих даних

Аналіз даних - це застосування до спеціальних даних алгоритмів фільтрації, сортування, агрегування, виконання арифметичних та статистичних операцій з метою виявлення закономірностей, прихованих тенденцій та ін. Таким чином, аналіз даних полягає в їх вибірці, агрегуванні або іншому перетворенні, що роблять інформацію зручною для вивчення. Справді, сирі дані, чи це великі дані, чи просто дані з файлів логів, людині важко прочитати. Вони є, по суті, таблиці або структуровані файли формату JSON/XML, і, просто переглядаючи їх, можна отримати надзвичайно мало інформації через психофізіологічні особливості людини.

Методи аналізу великих даних:

1. A/B тестування

Ця техніка аналізу даних передбачає порівняння контрольної групи з різними тестовими групами, щоб визначити, які способи лікування або зміни покращать задану об'єктивну змінну. McKinsey наводить приклад аналізу того, яка копія,

текст, зображення чи макет покращать коефіцієнт конверсії на сайті електронної комерції. Великі дані знову вписуються в цю модель, оскільки вона може тестувати величезні числа, однак цього можна досягти, лише якщо групи мають достатньо великий розмір, щоб отримати суттєві відмінності.

2. Злиття даних та інтеграція даних

Завдяки поєднанню набору методів, які аналізують та інтегрують дані з кількох джерел і рішень, ця інформація є ефективнішою та потенційно точнішою, ніж якщо б вона була розроблена з одного джерела даних.

3. Інтелектуальний аналіз даних

Поширений інструмент, який використовується в аналітиці великих даних, інтелектуальний аналіз даних витягує шаблони з великих наборів даних шляхом поєднання методів зі статистики та машинного навчання в рамках управління базою даних. Прикладом може бути, коли дані про клієнтів видобуваються, щоб визначити, які сегменти найімовірніше відреагують на пропозицію.

4. Машинне навчання

Добре відоме в області штучного інтелекту машинне навчання також використовується для аналізу даних. Виникаючи з інформатики, він працює з комп'ютерними алгоритмами, щоб створити припущення на основі даних. Він дає прогнози, які були б неможливі для людських аналітиків.

5. Обробка природної мови (Natural language processing – NLP, ОПМ). Відомий як підспеціальність інформатики, штучного інтелекту та лінгвістики, цей інструмент аналізу даних використовує алгоритми для аналізу людської (природної) мови.

6. Статистика. Ця техніка працює для збору, упорядкування та інтерпретації даних в рамках опитувань та експериментів.

Інші методи аналізу даних включають просторовий аналіз, прогнозне

моделювання, навчання правил асоціацій, мережевий аналіз та багато іншого.

Аналіз даних в хмарних середовищах

Аналіз даних в хмарних середовищах може бути наступних видів:

- потоковий — застосовується для аналізу потоків даних у реальному режимі або з невеликою затримкою;
- інтерактивний;
- пакетний - використовується для виконання періодичних завдань аналізу в сховищі даних, наприклад, агрегування, фільтрації та ін;
- інтелектуальний - це аналіз даних із застосуванням складних алгоритмів машинного навчання. Вжито термін «складних», оскільки у всіх У попередніх випадках алгоритм аналізу користувач пише сам. У разі ж інтелектуального аналізу цей алгоритм, а вірніше, його параметри визначаються шляхом процедури навчання безпосередньо на самих даних. Інтелектуальним цей аналіз називається з огляду на те, що в ньому застосовуються елементи штучного інтелекту, наприклад машинне навчання за допомогою алгоритмів на основі нейронних мереж.

Інтерактивний аналіз — служить для аналізу даних, що знаходяться в сховищі, активну участь користувача; останній при цьому вводить запит на веб-порталі або в терміналі та очікує відповідь протягом мінімального часу. Інтерактивний аналіз передбачає активну участь спеціаліста з обробки даних (data scientist). Ця людина висуває гіпотези, вибирає програмні інструменти аналізу, визначає джерела даних, застосовує до них запити мовою, яку підтримують ці інструменти. Сервіси інтерактивного аналізу надають засоби відображення інформації у вигляді таблиць чи графіків. Такий аналіз застосовується, коли треба вивчити дані у відповідь одноразовий запит бізнесу. Наприклад, встановити тенденцію в бізнес-даних, що не відображається у періодичних звітах. Проблема інтерактивного аналізу полягає в тому, що потрібно створити систему, яка дозволяє виконувати запити користувача до великого обсягу даних

(йдеться не про петабайти, а «лише» про багато гігабайтах) за досить малий час, що обчислюється хвилинами. Це необхідно для того, щоб робота спеціаліста з обробки даних була максимально ефективною і ланцюжок «висування гіпотез – написання запиту – виконання запиту - аналіз результату - коригування гіпотез...» займала найменше час. Очевидно, що для побудови такої системи підійдуть ресурси або послуги, що створюються на вимогу. Як правило, вони складаються із сховища даних, що допускає швидкий аналіз; кластера, що надає обчислювальні ресурси для побудови запиту та виконання паралельних обчислень; сервісів копіювання та трансформації даних, що служать для переміщення інформації із сторонніх джерел у це сховище.

Потоковий аналіз - полягає в аналізі потоку у вигляді послідовності повідомлень, тобто у застосуванні алгоритму аналізу до кожного повідомлення потоку. Цей аналіз може полягати у виділенні з усього потоку повідомлень, які відповідають певним ознакам, наприклад повідомлення про помилки, які мають бути направлені в окреме сховище або сервіс, що займається обробкою помилок. Наступний випадок — завдання маршрутизації повідомлень потоку в різні напрямки залежно від ознак, пов'язаних з повідомленням (наприклад, залежно від значення у певному полі повідомлення). Дуже цікавий приклад потокового аналізу повідомлень, що одночасно відноситься до інтелектуального, — аналіз банківських транзакцій та визначення в цьому потоці потенційно підозрілих (скажімо, видача великої суми у нетиповій
для платника країні або видача дрібних сум у різних банкоматах протягом
малого інтервалу часу та ін.).
По суті, потоковий аналіз цілком представлений як своєрідна фільтрація
вхідного потоку. Одиниця інформації, що надходить до системи потокового аналізу BigData, - повідомлення. Це порція інформації, що складається з низки полів, а саме як мінімум з ідентифікатора, тимчасової мітки та власне поля, що несе корисну інформацію. Крім того, можлива наявність інших полів: номери повідомлення в послідовності, У той же час класичні системи аналізу сигналів мають справу з періодичними відліками будь-якої величини, що

характеризуються тимчасовим становищем і власне величиною відліків.

На відміну від цього потоковий аналіз даних полягає в аналізі потоку векторних величин, оскільки кожен елемент даних можна представити векторною величиною, що складається із набору полів. Наступна відмінність – випадковий момент надходження повідомлення, що відрізняє його від строго періодичного потоку величин відліків у сигналу.

Пакетний аналіз. Він ще називається історичним, оскільки має на увазі аналіз історичних даних, тобто даних, накопичених протягом деякого часу роботи інформаційної системи. Аналіз історичних даних традиційно вважається історично (я навмисне застосував цю тавтологію) першою областю обробки великих даних. Суть його у тому, що алгоритми обробки застосовуються до даних, вже перебувають у сховищах. Самі алгоритми запускаються автоматично та не потребують інтерактивної участі людини. Сервіси AWS Data Pipeline та Azure Data Factory дозволяють запускати завдання аналізу даних у сховищах, і тому, строго кажучи, ми їх розглядали. У цьому розділі я наведу відомості, обійдені нашою увагою у попередніх розділах.

Таким чином, типова архітектура системи пакетного аналізу даних включає:

- сховище даних, що допускає виконання масивно-паралельних алгоритмів аналізу (як такий може виступати HDFS, DWH, реляційна БД та ін.). Очевидно, що для великих масивів це сховища на основі групи серверів - кластера;
- обчислювальні засоби, що забезпечують розпаралелювання аналітичного запиту та зведення результатів його виконання в єдиний вид знову-таки у вигляді кластера;
- алгоритм аналізу даних, що виконується у вигляді завдання в кластері обчислювальних засобів і представлений як виконуваний модуль на компілюваному мові чи сценарій (SQL, Python);
- сервіс управління підсистеми зберігання, обчислення та планування виконання завдання.

Традиційний засіб аналізу історичних даних - Hadoop, що є родоначальником практично всіх сучасних систем великих даних.

3.Методи та підходи для проведення аналізу емоційного забарвлення текстових даних

Методи емоційного аналізу текстових даних

Проблема розпізнавання емоцій у текстах чи повідомленнях є надзвичайно складною, але актуальною. Емоційні розрахунки використовуються у робототехніці, транспорті, ігровій індустрії, освіті, маркетингових дослідженнях, пошукових системах, людино-машинних інтерфейси та ін Дослідження в цій галузі дозволяють навчати штучний інтелект розпізнаванню людських емоцій та налаштовувати роботу різних інформаційних систем залежно від стану людини.

Складність визначення емоції у повідомленні обґрунтовується наступними причинами [10]:

1) Вираження емоцій залежить від контексту.

2) Частота слів важливіша за їх розташування.

3) Наявність різноманітних стилістичних прийомів.

Тема розпізнавання емоцій у повідомленнях досить велика і в її рамках виділяють кілька задач. Наведемо основні типи задач:

1) Аналіз тональності тексту – клас методів контент-аналізу комп'ютерна лінгвістика, призначена для автоматизованого виявлення в текстах емоційно забарвленої лексики та емоційної оцінки авторів (думок) стосовно об'єктів, про які йдеться в тексті [11].

Аналіз тональності тексту може знайти застосування у наступних областях:

− здатний допомогти навчити комп'ютер сприймати природний мова на рівні, наближеному до людського. Досі машина розуміла тексти на абстрактному рівні – в основному через лексеми (слова), які для неї мали форму (набір букв) і змістом (значення).

Ця концепція пропонує ввести ще одну функцію – лексичну тональність тексту;

− здатний значно підвищити якість машинного перекладу.

При перекладі не обійтися без первинного аналізу тексту та окремих слів – зокрема, аналізу тональності;

− наблизити спосіб мислення комп'ютера до людського, як деяка думка автора.

2) Аналіз суб'єктивності - визначити, чи є текст суб'єктивним чи об'єктивним, тобто. чи виражені в тексті емоції

3) Класифікація за емоціями – визначити, яка саме емоція виражена у тексті. Дослідження Пола Екмана показують, що використовуються 6 базових емоцій [12] – радість, злість, огида, страх, смуток та подив (Таблиця 1).

4) Вилучення думок – визначити, хто стосовно кого чи чого висловив у тексті думку, і яка тональність цієї думки. Грунтуючись на інструкціях Екмана по основних людських емоціям, можна створити коротку інструкцію емоцій.

Таблиця 1 – Приклади емоційно забарвлених речень.

Приклад емоції	**Емоція**
Я дуже рада за твої здобутки!	Радість
Я дуже розсерджений на тебе!	Злість
Як ти можеш це їсти?	Огида
Ця ситуація викликає у мене погане передчуття!	Страх
Мені дуже шкода…	Сум
Як і досі це працює?	Здивування

Текст, який аналізується, можливо будь-який – повідомлення, коментар, відгук тощо, можна охарактеризувати як прояв однієї або декількох з наступних базових емоцій:

− радість: відносять до станів, спричинених відчуттям задоволення. Ці стани варіюються від задоволення від допомоги іншим, теплого піднесеного почуття, яке люди відчувають, коли бачать доброту та співчуття, переживання легкості та задоволеності чи навіть насолоди нещастями іншої людини до радісної

гордості за свої досягнення чи переживання чогось дуже гарного та дивовижного;

− агресія: стани, спричинені почуттям неуспіху в нашому прогрес. Містить як роздратування, так і лють і варіюється від розчарування, яке є реакцією на неодноразові невдачі подолати перешкоду до гніву, спричиненого сильною незручністю, від суперечок до гіркоти - гніву після несправедливого поводження та мстивості;

− огида: для повідомлень, які виражають як ворожість, так і огиду. Вони варіюється від спонукання уникати чогось огидного або відрази від реакції на поганий смак, запах, річ або ідею;

− страх: свідчить про тривогу та жах. Стану варіюються від трепету - очікування можливої небезпеки, нервозності, до відчаю, реакції на нездатність зменшити небезпеку, паніки та жаху. Суміш страху, огиди та шоку;

− сум: містить одночасно розчарування та розпач. Стани варіюються від зневіри, безумства, безпорадності, безнадійності до сильних страждань, почуття та печалі, часто спричинених втратою або смутком та болем;

− здивування: коментарі, які виражають почуття, спричинені несподіваними подіями, у щось, у що важко повірити та які можуть вас шокувати. Це найкоротша емоція з усіх, вона триває за все декілька секунд. Це може бути страх, гнів, полегшення або нічим, залежно від події, яка нас дивує;

− інше: для текстів, які не показують жодної з перерахованих вище емоцій або не містять емоцій.

Перед вилученням емоцій із тексту необхідно його відфільтрувати. Повідомлення в більшості випадків зашумовані (наприклад, зайві або пропущені символи або літери, помилки правопису, смайлики, повторення знаків, хештеги та ін. Після «очищення» виходять відфільтровані чисті дані, з якими зручно працювати.

Підходи до вилучення емоцій із тексту.

Виділяють кілька підходів [13]:

1) Підхід, що базується на ключових словах. Найбільш інтуїтивний та сильний підхід. Базується на знаходженні шаблонів близьких до емоційним ключовим словам та зіставленні їх.

2) Підхід, заснований на лексиці. Класифікує текст використовуючи наявний лексикон (база знань текстів, помічених відповідно до емоціями) для вхідних даних.

3) Машинне навчання. Для розпізнавання емоцій використовуються навчання як з учителем, так і без, у яких модель спрямована на навчання та тестування класифікатора, при цьому дані поділяються на навчальну та перевірочну вибірку.

4) Змішаний підхід. Комбінує два або три методи для досягнення найкращого виграшу серед безлічі алгоритмів та найвищого рівня точності.

Для навчання алгоритмів машинного навчання потрібна вибірка. Вона потрібна для тренування моделі Machine Learning, щоб навчити систему, а потім використовувати її для вирішення реальних завдань [14].

Вибірку треба розбити на 7 класів емоцій: 1 – радість, 2 – агресивність, 3 – огида, 4 – страх, 5 – сум, 6 – подив, 7 – нейтральні.

Можна використовувати моделі машинного навчання, такі як:

1) Машина опорних векторів (SVM)

2) Випадковий ліс

І дві моделі глибокого навчання:

1) Згорткова нейронна мережа (CNN)

2) Рекурентна нейронна мережа (LSTM – «Довгострокова короткострокова пам'ять»).

Метод опорних векторів – це метод машинного навчання, призначений для вирішення завдань класифікації та регресії. Дослідники з OpenAI застосовували для розпізнавання емоцій LSTM та вид даної нейронної мережі добре

зарекомендував себе у розпізнаванні написаного тексту та людської мови. Їй належить рекорд мінімуму помилок під час розпізнавання мови – 17,7% [15].

Важлива частина машинного навчання це класифікація. Якщо потрібно знати, якого класу належить значення. Завдання класифікації добре вирішує випадковий ліс. Фундаментальна концепція в основі довільного лісу проста, але сильна - це знання більшості [16].

Для вирішення задачі автоматичного розпізнавання емоцій у аналізованої системі використовується згорткова нейронна мережа. Це особлива архітектура нейронних мереж, основним призначенням якої є ефективне розпізнавання образів [17].

Проблема автоматичного розпізнавання емоцій у текстах є нині дуже важливою та актуальною. Для її вирішення було розроблено безліч методів та алгоритмів. Всі вони мають свої перевагами та недоліками, і один із перспективних напрямів досліджень полягає у комбінуванні методів різних підходів.

Емоційний аналіз

Підхід на основі лексики(Lexicon based approach): підходи на основі лексики з використанням лексичних ознак [18] поділяються на два підвиди: на основі словника та на основі корпусу. Словникові підходи починаються з певного попередньо визначеного словника емоційно-забарвлених слів, а потім використовують різні метрики, як частота використання слова, кількість слів (так звана статистика) або синоніми слова (називаються семантикою) тощо для маркування речень у даних. Більшість статистичних підходів використовують латентний семантичний аналіз (LSA) і навіть їх варіації були використані для аналізу зв'язку між набором документів і словами у цих документах, щоб створити значущі моделі пов’язані з документами та словами. Словниковий підхід має низьку вартість, але в той же час перевірка словника може бути важкою.

Корпусний підхід використовує будь-які загальні дані для аналізу емоцій. Тут

спочатку анотується корпус (дані), дотримуючись набору абстрактних правил, з тексту, для управління аналізом емоцій природної мови. Підходи на основі ключового слова визначають набір заздалегідь визначених термінів для класифікації тексту за емоційним забарвленням. Страппарава використав WordNet-Affect [19] також для перевірки емоцій слів у заголовках.
Онтологічний підхід використовує зв'язок між термінами і EmotiNet [20] та моделює ситуації як ланцюг дій та їх відповідний емоційний вплив. Цей підхід є також застосовним для детального виявлення емоцій[21].

2. Підхід машинного навчання: спирається на алгоритми машинного навчання, які можуть навчатися за даними [22], використовуючи лінгвістичні особливості тексту.
Вони в свою чергу поділяються на наступні:
а) Контрольоване машинне навчання(Supervised Machine Learning): алгоритми формують функцію(модель) на основі вхідних даних і з використанням цієї функції приймає рішення про те, як співставити майбутні вхідні дані з належними вихідними даними [23][24]. SVM – це традиційний підхід в такому відношенні. Деякі дослідники [25] [26] вийшли за межі традиційних підходів до більш ефективних та надійних методів, як CRF [27].
Моделі навчання під наглядом:
Класифікатори дерева рішень: використовують ієрархічну рекурсивну декомпозицію навчальних даних на основі значеннь атрибутів; поки не буде досягнуто листкових вузлів, що містять значень, що підлягають класифікації [28];
Класифікатор на основі правил: класифікація моделюється на основі певного набору правил. Умови в диз'юнктивній нормальній формі розташовані у лівій частині правила, а мітки класів представлені правою стороною правила [29];
Лінійні класифікатори: класифікують емоції шляхом прийняття рішення на основі значення лінійної комбінації характеристик вхідного тексту. Ці характеристики також відомі як значення ознак і представлені у формі вектору (векторі ознак).

Імовірнісний класифікатор: передбачає, що кожен клас є компонентом суміші, що забезпечує ймовірність вибірки певного терміна для цього компонента. Крім того, він складається з наступних класифікаторів:
Наївний Байєсівський класифікатор, який обчислює апостеріорну ймовірність на основі розподілу слів у документі [30];
Байєсова мережа - ациклічний граф, вузли якого представляють випадкову величину і ребра представляють умовні залежності;
Максимальна ентропія, використовує кодування для перетворення позначених наборів об'єктів у вектори. Цей вектор потім обчислює ваги для кожної характеристики, які потім об'єднуються для визначення класу для кожного набору ознак [31][32].

б) Алгоритми машинного навчання без нагляду: намагаються знайти приховані структури у вхідних даних і використовують ці структури для співставлення немаркованих даних з класами емоцій [33].

в) Напівконтрольоване машинне навчання: існує багато робіт, де маркування виконується автоматично за допомогою хештегів чи інших засобів[34]. Напівконтрольовані алгоритми використовують цю ідею автоматичного маркування та застосовують наступні два підходи: Bootstrapping [35] та дистанційний нагляд [36][37].

Інфраструктурні (кластерні) рішення для аналізу інформації

Структура системи

Програма складається з трьох мікросервісів, які при одночасній роботі утворюють повноцінний, дієздатний і готовий до масштабування і розгортання у хмарних сервісах комплекс.
З технічної точки зору програма складається з трьох мікросервісів, кожен з яких вирішує певний набір завдань:

Frontend – веб-сервер Nginx, який обслуговує статичні файли React.

WebApp — веб-програма, написана на Java, яка обробляє запити від фронтенду.

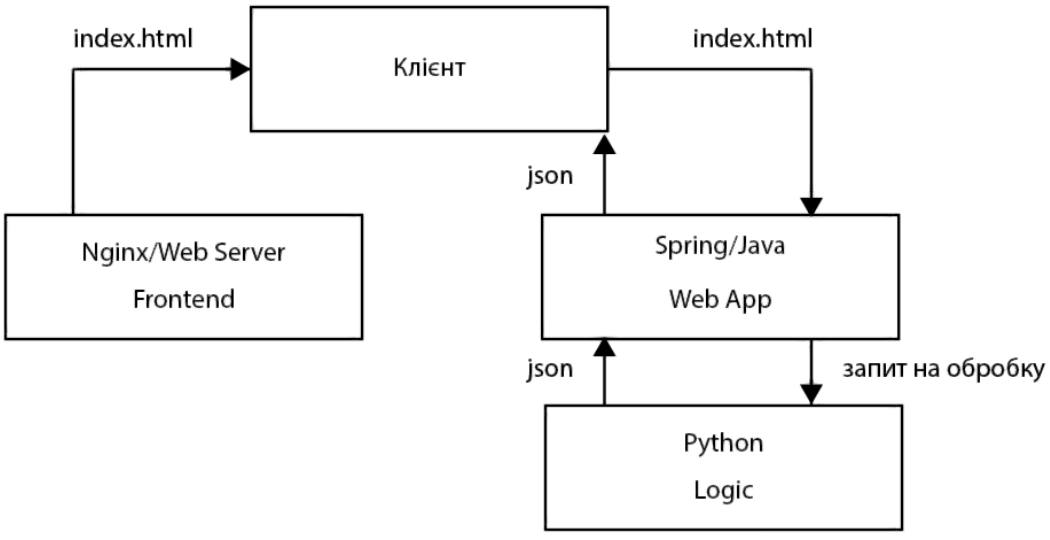
Logic — програма Python, яка виконує аналіз тональності тексту.

Мікросервіси існують не в ізоляції. Вони реалізують ідею «поділу обов'язків», але їм необхідно взаємодіяти один з одним.

Функціональний зв'язок компонентів

Принцип взаємодії мікросервісів виглядає наспуним чином:

- Браузер запитує у сервера файл index.html (який, у свою чергу, здійснює завантаження пакета React-програми).
- Користувач взаємодіє з програмою, це викликає звернення до веб-застосунку, заснованому на Spring.
- Веб-програма перенаправляє запит на виконання аналізу тексту Python-додатку.
- Python-додаток проводить аналіз тональності тексту та повертає результат у вигляді відповіді на запит.
- Spring-програма надсилає відповідь React-програмі (а вона, у свою чергу, показує результат аналізу тексту користувачеві).



Вимоги для розгортання

Для розгортання сервісу необхідно мати образи контейнерів кожного мікросервісу.

Образ контейнера - це легкий, автономний, виконуваний пакет, що містить певний додаток, який включає все необхідне для його запуску: код програми, середовище виконання, системні засоби та бібліотеки, налаштування.

Контейнеризованими програмами можна користуватися у середовищах Linux і Windows, вони завжди працюватимуть однаково незалежно від інфраструктури.

Для контейнеризації додатків у роботі було використано програмні засоби Docker. Для того, щоб отримати доступ до образів контейнерів через глобальну мережу, потрібно завантажити їх до репозиторію Docker Hub.

Для вирішення проблеми масштабування контейнеризованих застосунків використовуються програмні засоби Kubernetes.

Kubernetes — це система для автоматизації розгортання, масштабування та керування контейнерними програмами. Її ще називають "оркестратором контейнерів" (container orchestrator). Отже, необхідно встановити Minikube і Kubectl (клієнт, який дозволяє надсилати запити до API-сервера Kubernetes). У системі Kubernetes мінімальними обчислювальними одиницями, що розгортаються є поди («pod»).

Властивості подів:

- У кожного пода в кластері Kubernetes є унікальна IP-адреса.
- У поді може міститися безліч контейнерів. Вони спільно використовують доступні номери портів, тобто, наприклад, можуть обмінюватися один з одним інформацією через localhost (звісно, вони можуть користуватися одними й тими самими портами). Взаємодія з контейнерами, що знаходяться в інших подах, організується з використанням IP-адрес цих подів.
- Контейнери в подах спільно використовують томи сховища даних, IP-адресу, номери портів, простір імен IPC.

Далі необхідно створити і заповнити manifest file для кожного застосунку

окремо. У файлі опису пода мають бути зазначені наступні параметри, після чого поди будуть готові до запуску:

- Kind: визначає вид ресурсу Kubernetes, який ми хочемо створити.
- Name: Ім'я ресурсу.
- Spec: об'єкт, який визначає необхідний стан ресурсу. Найважливіша властивість тут – це масив контейнерів.
- Image: образ контейнера, який ми хочемо запустити у цьому поді.
- Name: унікальне ім'я контейнера, що знаходиться в поді.
- ContainerPort: порт, який прослуховує контейнер.

Наступним кроком аналогічно створюються файли опису сервісів. Сервіси Kubernetes грають роль точок доступу до наборів подів, які надають той самий функціонал, що й ці поди. Сервіси виконують вирішення завдань по роботі з подами та балансування навантаження між ними.

Розгортання (Deployment) – це абстракція Kubernetes, яка дозволяє нам керувати змінами у додатках. В циклі розробки додатку періодично змінюються вимоги до нього, розширюється його код, цей код упаковується і розгортається. При цьому на кожному кроці цього процесу можуть відбуватися помилки.

Ресурс виду Deployment дозволяє автоматизувати процес переходу від однієї версії до іншої. Це робиться без переривання роботи системи, а якщо в ході цього процесу станеться помилка, ми матимемо можливість швидко повернутися до попередньої, робочої версії програми. Після створення і заповнення файлу опису ресурсу Kubernetes виду Deployment система готова до розгортання. Розгортання відбувається шляхом виконання наступної послідовності дій:

-розгортання подів (приклад команди: kubectl apply -f sa-logic-deployment.yaml –record)

-застосування сервісів(приклад команди: kubectl apply -f service-sa-logic.yaml)

Результат роботи

Програма виконує лише одну функцію: вона приймає, як вхідні дані, одне речення англійською мовою, після чого, використовуючи засоби аналізу текстів, проводить аналіз тональності (sentiment analysis) цього речення, отримуючи оцінку емоційного ставлення автора речення до якогось об'єкта.

Приклад аналізу речення:

Речення “I love my motherland” отримало оцінку полярності 0.5, що свідчить про гарне ставлення автора речення до об’єкту мовлення.

4.Опис реалізованої аналітичної системи

Аналіз тональності тексту (англ. Sentiment analysis) – вид методів контент-аналізу в комп'ютерній лінгвістиці, призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті.

Тональність – це емоційне ставлення автора висловлювання до деякого об'єкту, виражене в тексті. Тональність всього тексту в цілому можна визначити як функцію (в найпростішому випадку суму) лексичних тональностей складових його одиниць (речень) і правил їх поєднання.

У сучасних системах автоматичного визначення емоційної оцінки тексту найчастіше використовується одновимірний емотивний простір: позитив чи негатив (добре або погано).

Усі методи сентимент-аналізу текстового контенту можна класифікувати за декількома ознаками:

1) За шкалою оцінювання:
   a) Бінарна шкала
   b) Системи шкалювання
   c) Суб’єктивність/об’єктивність
2) За автоматизацією процесу:
   a) Ручний(аналіз тональності експертами)
   b) Автоматизований
3) За методикою оцінювання:
   a) Методи, засновані на правилах або словниках
   b) Методи машинного навчання без вчителя
   c) Методи машинного навчання з вчителем
   d) Методи, засновані на теоретико-графових моделях

Згідно з наданою класифікацією, у даній роботі використано автоматизований метод, заснований на словнику із поданням результату за системою шкалювання (-1;1).

5. Висновки

В кваліфікаційній роботі досліджено та проаналізовано способи отримання великих наборів даних, основні принципи роботи з ними, методи накопичення та збереження, підходи до їх обробки. Розглянуто принципи побудови та види аналітичних систем для роботи з великими даними, визначено труднощі, які виникають під час їх розробки (масштабування, слабка структурованість даних, недостатня визначена обсягів ресурсів для виконання задач аналізу великих даних). Запропоновано використання хмарних обчислень. Розглянуто основні засади класичної архітектури хмарних обчислень. Досліджено та вирішено задачу емоційного аналізу слабко структурованих текстових даних, деталізовано методи її розв’язання.

Для дослідження текстового забарвлення (ставлення авторів тексту до об’єкту обговорення) в запропонованій аналітичній системі використовується словниковий підхід. Забезпечення масштабування: реалізацію проведено за допомогою використання хмарних сервісів та технологій. Визначено сфери людської діяльності, у яких доречно застосувати розроблену систему.

Для забезпечення зручності у використанні системи створено інтерактивний веб-портал, через який забезпечується доступ до реалізованої системи, викладено принципи її підготовки та розгортання на будь-якій інфраструктурі та у будь-яких масштабах.

Отримано висновок, що запропоноване рішення виявилося ефективним на фоні можливості асинхронного способу внесення змін у системі без перебоїв у роботі сервісів та в умовах оперативного корегування об’ємів задіяних ресурсів.

Список використаних джерел

1. Big Data in Big Companies [Електронний ресурс]. : http://www.datascienceassn.org/sites/default/files/Big%20Data%20in%20Big%20Companies%20-%20Tom%20Davenport.pdf
2.Frank A. Ohlhorst Cloudy Year for Big Data. eWeek [Electronic Resours] / Frank A. – Access mode: http://www.eweek.com/c/a/Cloud-Computing/ 2012-A-Cloudy-Year-for-Big-Data-102807.
3. The Open Archives Initiative Protocol for Metadata Harvesting Protocol Version 2.0 of 2002-06-14 eWeek [Electronic Resours]. – Access mode: http://www.openarchives.org/OAI/2.0 /openarchivesprotocol.htm.
4. Laney D. The Importance of «Big Data»: A Definition [Text] [Electronic Resours] / Mark A. Beyer, Douglas Laney. – Access mode: https://www.gartner.com/doc/2057415/importance-big-data-definition.
5. Шаховська Н. Б. Організація великих даних у розподіленому середовищі / Н. Б. Шаховська, Ю. Я. Болюбаш, О. М. Верес // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. – 2014. –№ 2. – С. 147–155.
6. Шаховська Н. Б. Робота з великими даними показниками соціоеколого-економічного розвитку регіону / Н. Б. Шаховська, Ю. Я. Болюбаш // Восточно-Европейский журнал передовых технологий. – 2013. – № 5(2). – С. 4–8.
7. Jacobs A. The Pathologies of Big Data [Text] / Jacobs A. // Databases. – 2009. – Vol. 7, issue 6. – P.1-12.
8. Magoulas R. Introduction to Big Data [Electronic Resource] / R. Magoulas, B.Lorica. – Access mode: http://www.oreilly.com/data/free/release-2-issue11.csp.
9. Digital universe of opportunities [Електронний ресурс]. – Режим доступу: http://www.emc.com/leadership/digital-universe/index.htm?pid=landing-digitaluniverse-131212.
10. Review of approaches for automatic recognition of emotions in texts [electronic resource] // https://cyberleninka.ru/article/n/obzor-podhodov-dlyaavtomaticheskogo-raspoznavaniya-emotsiy-v-tekstah
11. Analysis of the sentiment of the text: concept, methods, areas of application [electronic resource] // http://datareview.info/article/analiz-tonalnosti-tekstakontseptsiya-metodyi-oblasti-primeneniya/

12. Wikipedia site: star-wiki.ru - Emotion recognition [electronic resource] // https://star-wiki.ru/wiki/Emotion_recognition#Emotion_recognition_in_text

13.Using modern machine learning algorithms for the emotion recognition problem [electronic resource] // https://cyberleninka.ru/article/n/ispolzovaniesovremennyh-algoritmov-mashinnogo-obucheniya-dlya-zadachiraspoznavaniya-emotsiy/viewer

14. Let's select what you need Data Mining: how to form a dataset for machine learning [electronic resource] // https://www.bigdataschool.ru/blog/datasetdata-preparation.html

15. Artificial intelligence was taught to better recognize emotions in the text [electronic resource] // https://aboutdata.ru/2017/04/08/ai-sentimentrecognition/

16. How does the random forest work? [electronic resource] // https://nuancesprog.ru/p/6160/

17. The use of neural network technologies in the problem of automatic recognition of emotions / SO Tselikova, Ya. P. Gorozhankin, AO Ivanov [and others]. - Text: direct // Young scientist. - 2019. - No. 26 (264). - S. 59-61. - URL: https://moluch.ru/archive/264/61173

18. Carlo Strapparava and Rada Mihalcea- Learning to identify emotions in text. In Proceedings of the 2008 ACM Symposium on Applied Computing, SAC ’08, pages 1556–1560, New York, NY, USA. ACM,2008.

19. Carlo Strapparava and Alessandro Valitutti -WordNet-Affect: an Affective Extension of Word-Net. In 4th International Conference on Language Resources and Evaluation, pages 1083–1086, (2004).

20. Balahur A, Hermida JM, Montoyo A, Munoz R- Emotinet: A Knowledge Base for Emotion Detection in Text Built on the Appraisal Theories- In: Munoz R, Montoyo A, Metais E (eds) Natural Language Processing and Information System. NLDB 2011. Lecture Notes in Computer Science, vol 671, (2011) - Springer

21. Martin D Sykora, Thomas W Jackson, and Suzanne Elayan- Emotive ontology: extracting fine-grained emotions from terse, informal messages. IADIS International Journal on Computer Science and Information Systems, 8(2):106–118,(2013).

22. Ron Kovahi and Foster Provost- Glossary of terms. Machine Learning, pages 271–274, (1998).

23. C. M. Bishop- Pattern Recognition and Machine Learning,(2006), Springer.

24. Mehryar Mohri, Afshin Rostamizadeh, and Ameet TalwalkarFoundations of Machine Learning.MIT Press,(2012).

25. Jerome R Bellegarda- Emotion analysis using latent affective folding and embedding. In Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text, pages 1–9. Association for Computational Linguistics,2010.

26. Xuren Wang and Qiuhui Zheng- Text Emotion Classification Research Based on Improved Latent Semantic Analysis Algorithm. In Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), number Iccsee, pages 210–213, Paris, France. Atlantis Press,2013.

27. Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-His Chen- Emotion classification using web blog corpora. In Web Intelligence, IEEE/WIC/ACM International Conference on, pages 275–278. IEEE,2007.

28. J.R. Quinlan Induction of decision trees Machine Learn, 1 pp. 81– 106,(1986).

29. W.Medhat, A Hassan, H Korashy-combined algorithm for data mining using association rules; Ain Shams J Electrical Eng, 1(1), (2008).

30. H Kang, SJ Yoo, D Han - Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews: Expert Systems with Applications, (2012) – Elsevier

31. M Kaufmann - J MaxAlign: A Maximum Entropy Parallel Sentence Alignment Tool. - COLING (Demos), (2012).

32. Diman Ghazi, Diana Inkpen, and Stan Szpakowicz- Hierarchical versus flat classification of emotions in text. In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, CAAGET ’10, pages 140–146, Stroudsburg, PA, USA. Association for Computational Linguistics,2010.

33. A Agrawal, A An- Unsupervised emotion detection from text using semantic and syntactic relations: - Proceedings of The 2012 IEEE/WIC/ACM International joint conference on web intelligence and intelligent Agent technology, volume 1, pages 346- 353,(2012).

34. JD Rodriguez, L Alzate, M Lucania, I Inza, JA Lozano – Approaching Sentiment Analysis by using semi-supervised learning of multidimensional classifiers, Neurocomputing, (2012) – Elsevier

35. L Canales, C Strapparava, E Boldrini, P Martnez-Barco; Exploiting a Bootstrapping Approach for Automatic Annotation of Emotions in Texts: (2016) IEEE.

36. Jared Suttles and Nancy Ide- Distant Supervision for Emotion Classification with Discrete Binary Values. In Alexander Gelbukh, editor, Computational Linguistics and Intelligent Text Processing, volume 7817 of Lecture Notes in Computer Science,pages 121– 136,(2013) Springer Berlin Heidelberg

37. Purver, M., Battersby, S.- Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 482{491. Association for Computational Linguistics, Avignon, France, 2012.