НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА АНАЛІЗУ ДАНИХ

«До захисту допущено»

В. о. завідувача кафедри

_____ І. М. Терещенко

(підпис) (ініціали, прізвище)

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 113 «Прикладна математика»

(код і назва)

на тему: **Крос-модальні представлення налаштовані на MusicCaps для пошуку музики за текстовим описом**

Виконав: студент 4 курсу, групи ФІ-92

Плахтій Гліб Олексійович _____

(прізвище, ім'я, по батькові) (підпис)

Керівник асистент кафедри ММАД Яворський О. А. _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____

(номер розділу) (посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент старший викладач кафедри ІБ Наконечна Ю. В. _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2023

Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий Фізико-технічний інститут
Кафедра математичного моделювання та аналізу даних

Рівень вищої освіти — перший (бакалаврський)
Спеціальність 113 «Прикладна математика»

«ЗАТВЕРДЖЕНО»
В. о. завідувача кафедри
_____ І. М. Терещенко
(підпис) (ініціали, прізвище)
«___» _____ 2023 р.

ЗАВДАННЯ
на бакалаврську роботу студенту

Плахтію Глібу Олексійовичу _____
(прізвище, ім'я, по батькові) (підпис)

1. Тема роботи: Крос-модальні представлення налаштовані на MusicCaps для пошуку музики за текстовим описом,
науковий керівник роботи асистент кафедри ММАД Яворський О. А.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «_____»_____2023 р №

2. Термін подання студентом роботи «_____»_____2023 р.

3. Об'єкт дослідження: Моделі для пошуку музики за текстовим описом які використовують крос-модальні представлення
4. Предмет дослідження: Моделі машинного навчання для отримання крос-модальних представлень текстових описів та музики
5. Перелік завдань, які потрібно розробити: Зібрати датасет який складається з музики та опису до неї; реалізувати різні модифікації моделі та протестувати для різних задач; провести тестування моделей від сторонніх розробників на різних задачах та порівняти з розробленою.
6. Орієнтовний перелік ілюстративного матеріалу: презентація — 14 аркушів А4

7. Дата видачі завдання «__15__»_____січня_____2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1.	Опрацювання літератури за темою	01.02.2023 - 20.02.2023	Виконано
2.	Збір та обробка датасету	01.03.2023 - 05.03.2023	Виконано
3.	Реалізація та тестування різних архітектур моделі	20.03.2023 - 01.05.2023	Виконано
4.	Написання першого та другого розділів дипломної роботи	01.05.2023 - 20.05.2023	Виконано
5.	Тестування моделей у відкритому доступі	20.05.2023 - 25.05.2023	Виконано
6.	Написання третього розділу дипломної роботи	25.05.2023 - 31.05.2023	Виконано
7.	Написання висновків	01.06.2023 - 04.06.2023	Виконано
8.	Підготовка презентації на передзахист	04.06.2023 - 07.06.2023	Виконано

Студент _____ Г. О. Плахтій
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ О. А. Яворський
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Пояснювальна записка дипломної роботи за обсягом становить 42 сторінки, містить 12 таблиці та 8 рисунків. Для дослідження було використано 11 бібліографічних найменувань.

Музика є важливою складовою нашого життя. І з кожним днем її кількість тільки збільшується. Тому проблема пошуку музики є дуже актуальною.

Текстовий опис є одним з основних способів, за допомогою якого люди виражають свої музичні вподобання або шукають певні типи пісень. Наприклад, користувачі можуть використовувати слова, які описують настрій (веселий, сумний, енергійний), жанр (рок, поп, електронна музика) або характеристики звучання (акустичний, експериментальний, ритмічний) для пошуку музики, яка відповідає їхнім потребам.

В даній роботі для вирішення цієї задачі використовується моделі машинного навчання для створення крос-модальних представлень. Дана модель створює такі векторні представлення музики та тексту, що знаходяться близько один до одного, якщо текст описує музику. За допомогою цього з бібліотеки музики можна вибрати ті треки які найбільше підходять під текстовий опис.

Основний внесок цієї дипломної роботи полягає в використані унікального датасетсу в межах задачі пошуку музики за текстовим описом. Також в цій роботі були запропоновані нові модифікації існуючих моделей.

Ключові слова: *машинне навчання, крос-модальні представлення, MusicCaps, трансформери, пошук музики за текстовим описом.*

SUMMARY

The diploma work explanatory note includes 42 pages of the text, 12 tables and 8 illustrations. At the problem modern state analysis, overall 11 references were used.

Music is an important part of our life. And every day its number only increases. Therefore, the problem of music retrieval is very relevant.

Text description is one of the main ways people express their music preferences or search for specific types of songs. For example, users can use words that describe mood (happy, sad, energetic), genre (rock, pop, electronic music), or sound characteristics (acoustic, experimental, rhythmic) to find music that fits their needs.

This work uses machine learning models to create cross-modal embeddings to solve this problem. This model creates vector embeddings of music and text such that the embeddings of music and text are close to each other if the text describes the music. With this, you can select from the music library those tracks that are most suitable for the text description.

The main contribution of this thesis is the use of a unique dataset for model training. New ones were also proposed in this work modifications of existing models.

Keywords: *machine learning*, *cross-modal embeddings*, *MusicCaps*, *transformers*, *text-to-music retrieval*.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ММН — модель машинного навчання

КМП — крос-модальні представлення

BERT — Bidirectional Encoder Representations from Transformers модель архітектури трансформер для обробки природної мови

AST — Audio Spectrogram Transformer модель архітектури Трансформер для обробки аудіо

NLP — Natural language processing обробка природної мови

ВСТУП

Актуальність дослідження. Проблема пошуку музики за текстовим описом є актуальною в сучасному світі, оскільки музична індустрія є однією з найбільш динамічних та впливових галузей розвитку культури і розваг у світі. Вона охоплює широкий спектр діяльності. За останні роки музична індустрія пройшла великі зміни через технологічний прогрес та вплив Інтернету. Цифрові технології дозволили легше записувати, продюсувати і поширювати музику. Через це з кожним днем кількість музики тільки збільшується. Тому постає проблема з пошуком музики. У світі, де доступно величезне розмаїття музичних жанрів, виконавців і пісень, багато людей шукають нову музику, що відповідає їхнім смакам і настрою. Традиційні методи пошуку, такі як використання назви виконавця або пісні, можуть бути обмежені і не завжди дають очікувані результати. Саме тому пошук музики за текстовим описом є дуже актуальним. Текстовий опис є одним з основних способів, за допомогою якого люди виражають свої музичні вподобання або шукають певні типи пісень.

Застосуванням технологій штучного інтелекту і машинного навчання в цій галузі можна покращити пошукові системи та рекомендаційні алгоритми, що дозволить людям знаходити музику, яка відповідає їхнім унікальним смакам та настрою. Це може допомогти зробити процес пошуку музики більш зручним для користувачів. Крім того, пошук музики за текстовим описом має потенціал для використання в різних сферах, таких як музична індустрія, реклама, кіно і телебачення. Музичні продюсери, рекламні агентства та інші зацікавлені сторони можуть використовувати отриману модель. Тому виникає потреба в створенні моделі яку можна використовувати для пошуку музики за текстовим описом. Вирішенням цієї задачі може бути модель для отримання крос-модальних представлень (КМП) музики та тексту. Цю модель машинного навчання (ММН) можна використовувати для вирішення багатьох задач.

Метою дослідження є розроблення моделі машинного навчання здатної працювати з крос-модільними даними (такими як пари текст-музика), відповідно до попередньо заданих метрик.

Завданням дослідження є розробка та тестування різних архітектур моделей та визначення найкращої. Також порівняння отриманих моделей з моделями від сторонніх розробників. Для цього необхідно: зібрати датасет який складається з музики та опису до неї; реалізувати різні модифікації моделі та протестувати для різних задач; провести тестування моделей від стороніх розробників на різних задачах та порівняти з розробленою.

Об'єктом дослідження є моделі для пошуку музики за текстовим описом які використовують крос-модальні представлення.

Предметом дослідження є моделі машинного навчання для отримання крос-модальних представлень текстових описів та музики.

Методи дослідження: методи математичної статистики, математичного аналізу та методи машинного навчання.

Наукова новизна полягає в використані унікального датасетсу в межах задачі пошуку музики за текстовим описом. Також в цій роботі були запропонованні різні підходи до розморозки претеренованих шарів енкодерів та модифікації моделі з додаванням додаткових шарів уваги до претренованих енкодерів.

Практичне значення полягає в отримані моделі яку можна використовувати для різних задач, таких як пошук музики за описом. Отримана модель має потенціал для використання в різних сферах, таких як музична індустрія, реклама, кіно і телебачення. Музичні продюсери, рекламні агентства та інші зацікавлені сторони можуть використовувати отриману модель. Дана модель після допрацювання буде використана в організації HARMIX INC. для надання користувачам можливості пошуку музики за описом.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1. Основи машинного навчання

Машинне навчання — це галузь штучного інтелекту, яка займається розробкою алгоритмів та моделей, що дають розробленим системам здатність навчатися та вдосконалювати свою продуктивність на основі даних. Вона використовує статистичні методи для навчання комп'ютерів розпізнавати патерни, здійснювати прогнози, приймати рішення та виконувати завдання без явного задання правил. Центральною концепцією машинного навчання є здатність алгоритмів «вчитися» на основі даних, виявляти тенденції, залежності та складні взаємозв'язки.

Машинне навчання використовується в багатьох галузях, включаючи комп'ютерне бачення, обробка природної мови, рекомендаційні системи, фінансовий аналіз, медицину та багато інших. Алгоритми машинного навчання можуть класифікувати дані, здійснювати кластеризацію, виконувати регресійний аналіз, генерувати прогнози та робити висновки на основі великого обсягу даних.

Для успішного використання машинного навчання необхідні відповідні дані, потужні алгоритми та потужний обчислювальний ресурс. Важливим етапом є підготовка даних, вибір підходящих моделей та налаштування параметрів. Моделі машинного навчання є основними інструментами для вирішення завдань у сфері штучного інтелекту. Вони використовуються для виявлення патернів у вхідних даних, здійснення прогнозів, класифікації, кластеризації, генерації та багатьох інших завдань. Існує багато різних архітектур моделей машинного навчання, кожна з яких має свої особливості та застосування. Однією з таких є нейронні мережі (neural network).

Нейронні мережі - це комп'ютерні моделі, які взяли за основу структуру та функціонування нейронної системи людського мозку. Вони призначені для вирішення завдань обробки і аналізу даних. Основною одиницею нейронної мережі є нейрон. Нейрони об'єднані в шари, інформація передається від одного шару до іншого через зв'язки між нейронами. Кожен нейрон отримує вхідні сигнали, обробляє їх та передає результат наступним нейронам.

Нейронні мережі навчаються шляхом використання алгоритмів, які оптимізують параметри мережі, щоб забезпечити оптимальні результати для певної задачі. Цей процес називається навчанням з учителем, коли для навчання використовується набір даних з позначеними правильними відповідями.

Існує багато архітектур нейронних мереж які використовуються для різних задач, наприклад Нейронна мережа прямого поширення (Feedforward Neural Networks) є найпростішим типом нейронних мереж. Інформація переміщується в одному напрямку, від вхідного шару до вихідного шару, без циклічних зв'язків. Кожен нейрон у шарі отримує сигнали від нейронів попереднього шару, обробляє їх і передає результат наступному шару. НМПП використовується для різних завдань, включаючи класифікацію, регресію та розпізнавання образів.

Рекурентна нейронна мережа (Recurrent Neural Networks) використовується для обробки послідовних даних, таких як мовний текст або часові ряди. Вона має циклічні зв'язки, які дозволяють передавати попередній вихідний стан як вхідний для наступного кроку обробки. Це дозволяє РНМ моделювати залежності в часі і розуміти контекстуальну інформацію. Модифікаціями РНМ є LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit). Використовуються в машинному перекладі, аналізі тексту, генерації мови та інших задачах, де використовується послідовність.

Трансформер (Transformer) - це архітектура нейронної мережі, яка здобула значний успіх у завданнях обробки природної мови, зокрема в машинному перекладі. Вона використовує механізми уваги для моделювання взаємодії між словами у вхідному тексті. Трансформер не має рекурентних або згорткових

шарів і може обробляти всі слова у вхідному тексті паралельно, що робить його високоефективним. Це дозволяє залучати більше даних для навчання та покращує якість обробки мови. Трансформер також знайшов застосування у генерації тексту, аналізі емоцій, розпізнаванні мови та багатьох інших задачах обробки природної мови.

Важливими компонентами нейронних мереж є функції активації та функція втрат. Функції активації (activation function) в нейронних мережах використовуються для введення нелінійності в вихідні дані нейронів і забезпечення необхідної гнучкості моделі. Вони приймають вхідний сигнал і генерують вихідний сигнал, який передається наступним нейронам у мережі. Найпоширеніші функції активації включають:

Сигмоїд (Sigmoid) - ця функція приймає будь-яке число і перетворює його на значення між 0 і 1. Вона часто використовується в задачах бінарної класифікації, де необхідно вирішити, чи належить вхідний зразок до певного класу чи ні.

$$S(x) = \frac{1}{1 + e^{-x}}$$

ReLU (Rectified Linear Unit) - ця функція встановлює вихідний сигнал рівним нулю, якщо вхідний сигнал менше нуля, і залишає його без змін, якщо вхідний сигнал більше нуля. ReLU широко використовується в глибоких нейронних мережах через свою ефективність та здатність підвищувати швидкість навчання.

$$ReLU(x) = \max(x, 0)$$

Softmax - ця функція, яка використовується в останньому шарі нейронної мережі для задачі багатокласової класифікації. Вона приймає вхідний вектор і виводить ймовірності належності вхідного зразка до кожного класу.

$$\sigma(\mathbf{z})_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{K} e^{\mathbf{z}_j}}, \text{ для } i = 1, ..., K \text{ і } \mathbf{z} = (z_1, ..., z_K) \in \mathbb{R}^K$$

Функція втрат (loss function) використовується для оцінки різниці між прогнозованими значеннями моделі і правильними відповідями у процесі навчання. Ця функція вимірює величину помилки та допомагає підлаштувати параметри мережі, щоб зменшити цю помилку

Найпоширеніші функції втрат включають:
Mean Squared Error (MSE, середньоквадратична помилка) - ця функція обчислює середнє значення квадратів різниці між прогнозованими і правильними значеннями. Вона часто використовується в задачах регресії.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

де Y_i: істине значення, а $\hat{Y}_i$: передбачене значення

Cross-Entropy Loss (втрата перехресної ентропії) - ця функція використовується в задачах класифікації, де потрібно оцінити розбіжність між прогнозованою ймовірністю класів та фактичними значеннями. Вона широко застосовується, наприклад, в задачах багатокласової класифікації.

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i)$$

де t_i : істине значення, p_i : ймовірність належності до класу і

Binary Cross-Entropy Loss (втрата перехресної ентропії для бінарної класифікації) - ця функція використовується в задачах бінарної класифікації, де потрібно оцінити розбіжність між прогнозованою ймовірністю двох класів та фактичними значеннями.

$$L = t \log p + (1 - t) \log (1 - p)$$

де t: істинне значення, p: ймовірність належності до класу 1

Узагальнюючи, машинне навчання — це важлива галузь, яка дає моделям здатність навчатися та виконувати завдання на основі даних. Вона має потенціал трансформувати різні сфери життя та принести значні переваги в сучасному світі.

1.2. Моделі архітектури Трансформер

Трансформер є одним з найбільш впливових архітектур в галузі обробки природної мови та машинного перекладу. Вперше запропонований у 2017 році в статті «Attention Is All You Need» [1] від команди дослідників з Google Brain, Трансформер змінив підхід до розв'язання завдань, пов'язаних з послідовними даними.

Основною ідеєю Трансформера є використання механізму уваги (attention mechanism), який дозволяє моделі зосереджуватися на різних частинах вхідних даних при обробці послідовностей. У традиційних послідовних моделях, таких як рекурентні нейронні мережі (RNN), інформація передається через покрокові обчислення, що є доволі повільним. Трансформер робить це шляхом використання механізму уваги для безпосередньої взаємодії між різними частинами вхідних даних.

Архітектура Трансформера (Рис. 1.1) складається з двох основних компонентів: енкодера і декодер. Енкодер приймає на вхід послідовність символів і створює контекстний вектор (сталої довжини) для кожного символу. Декодер також приймає послідовність символів, але він має доступ до контекстних векторів, згенерованих енкодером, і використовує їх для генерації вихідної послідовності.

Енкодер та декодер в Трансформері складається з множини шарів. Кожен шар має два підрівні: механізм уваги та позиційно-лінійна зв'язність (positional feed-forward network). Механізм уваги дозволяє моделі фокусуватися на різних

частинах послідовності, враховуючи важливість кожного символу в контексті інших символів. Позиційно-лінійна зв'язність відповідає за локальне зміщення при обчисленні шару. Шари енкодера використовуються для обробки вхідної послідовності, а шари декодера - для генерації вихідної послідовності.

Одна з головних переваг Трансформера полягає в його здатності до паралельного обчислення, оскільки кожен токен може бути оброблений незалежно. Це робить архітектуру Трансформера більш ефективною у порівнянні з традиційними послідовними моделями, такими як RNN.

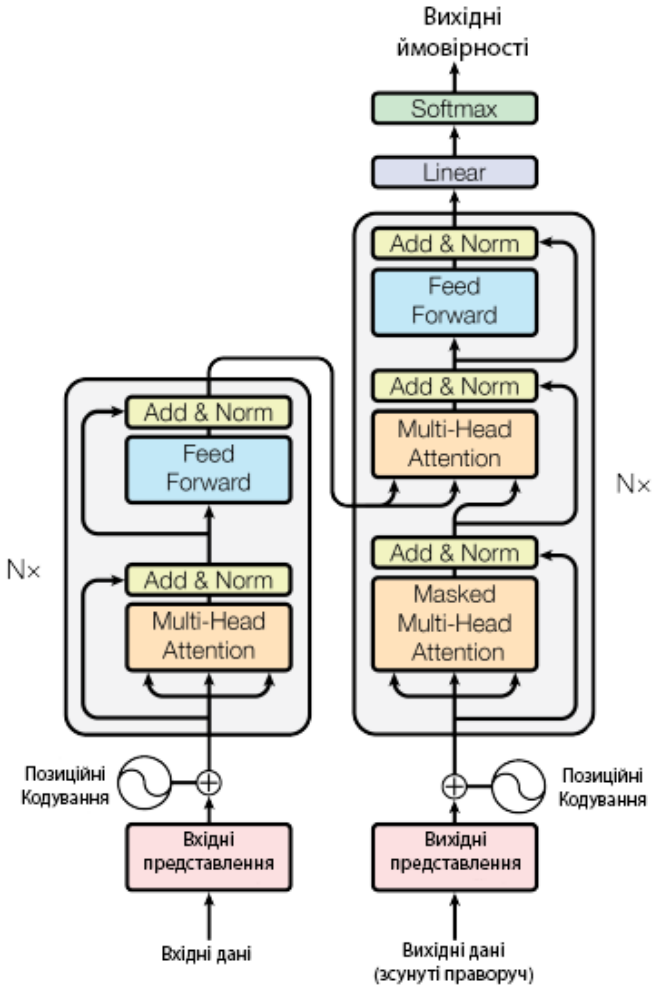

Рис. 1.1. Архітектура моделі типу трасформер [1]

Трансформер використовується в багатьох завданнях обробки природної мови, таких як машинний переклад, розпізнавання мови, генерація тексту та інші. Його успіх свідчить про значний прорив у галузі моделювання по-

слідовностей та продемонстрував важливість механізму уваги при роботі з послідовностями даних. Архітектура Трансформера продовжує бути предметом активних досліджень та вдосконалення, що сприяє подальшому розвитку сфери обробки природної мови та машинного перекладу.

Механізм уваги (attention mechanism) є ключовим компонентом архітектури Трансформера і грає важливу роль у розумінні та генерації послідовностей даних. Вперше запропонований у статті [1] , механізм уваги дозволяє моделі фокусуватися на різних частинах вхідних даних залежно від їх важливості для вирішення конкретного завдання. Основна ідея механізму уваги полягає в тому, що кожен символ або елемент вхідної послідовності отримує вагу (вектор уваги), яка вказує, наскільки важливим є цей символ для обробки поточного символу або генерації вихідного символу. Ваги визначаються шляхом обчислення скалярного добутку між вектором запиту (залежний від поточного символу) і векторами ключів та значень, які відповідають різним символам вхідної послідовності (Рис.1.2).

Основний процес механізму уваги можна розділити на кілька кроків:

1) Генерація запиту: Для кожного символу у процесі обробки або генерації використовується вектор запиту, який визначає, на що модель має зосередитися.

2) Обчислення ключів та значень: Для кожного символу вхідної послідовності обчислюються вектори ключів і вектори значень. Вектори ключів використовуються для визначення важливості символу, а вектори значень містять контекстну інформацію, яка пов'язана з кожним символом.

3) Обчислення скалярного добутку: Обчислюється скалярний добуток між вектором запиту і векторами ключів. Це визначає ступінь взаємодії між символами.

4) Обчислення ваг: Скалярні добутки нормалізуються за допомогою функції активації, такої як Softmax, для отримання ваг, які в сумі дають 1. Ці ваги вказують, яку увагу модель має приділити кожному символу.

5) Вагова сума значень: За допомогою обчислених ваг виконується взважена сума векторів значень, що відповідають символам вхідної послідовності. Це дає контекстний вектор, який представляє зосереджену інформацію залежно від важливості символів.

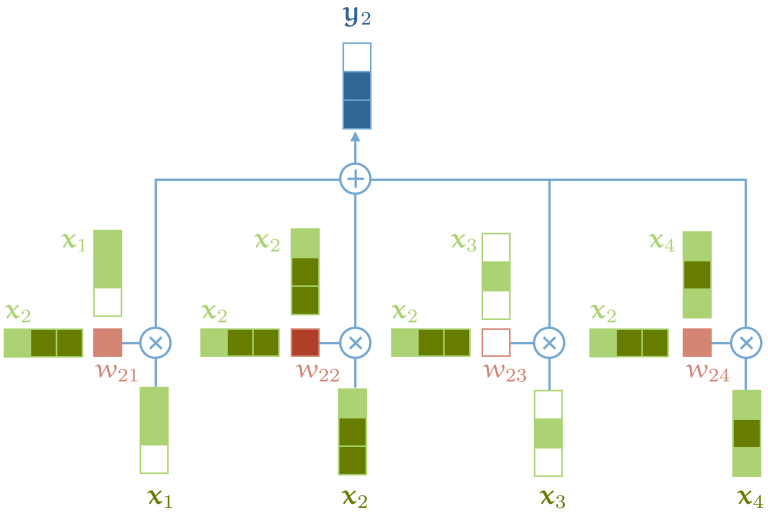

Рис. 1.2. Механізм уваги [2]

Механізм уваги дозволяє моделі зосереджуватися на важливих аспектах вхідних даних, що покращує якість перекладу, генерації тексту та інших задач обробки природної мови. Крім того, механізм уваги також покращує здатність моделі до паралельного обчислення, оскільки ваги можуть бути обчислені незалежно для кожного символу. Це забезпечує ефективну обробку послідовностей в Трансформері та інших моделях, які використовують механізм уваги.

Першою складовою архітектури моделі для навчання КМП є модель BERT(Bidirectional Encoder Representations from Transformers) [3] є однією з найвпливовіших архітектур в галузі обробки природної мови (NLP). Вперше представлений в 2018 році командою дослідників з Google, BERT використовує архітектуру Трансформера і досягає вражаючих результатів в багатьох завданнях, включаючи машинний переклад, розпізнавання іменованих сутностей, синтаксичний розбір та інші.

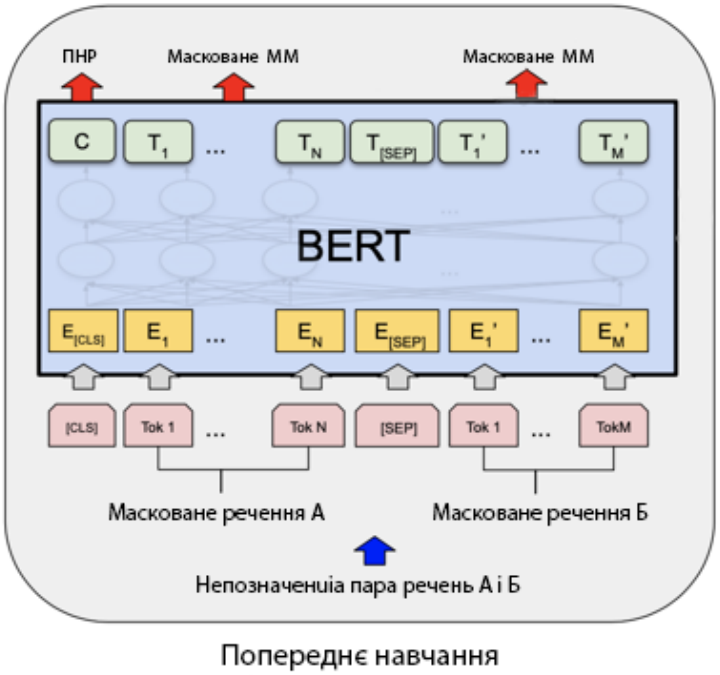

Рис. 1.3. Попереднє навчання BERT [3]

Під час навчання BERT вчиться передбачати масковані слова в реченні, заміняючи їх спеціальними токенами [MASK] (Рис. 1.3). Це дозволяє моделі отримати контекстуальні представлення слів, що базуються на контексті їхнього вживання в реченні. Також BERT навчається передбачати, чи наступне речення є логічним продовженням поточного (див. Рис. 1.3). BERT досягає вражаючих результатів завдяки використанню глибоких багатошарових нейронних мереж, основаних на архітектурі Трансформера. Модель має велику кількість параметрів і здатна ефективно захоплювати складні залежності в послідовностях даних. Крім того, BERT використовує механізм уваги, який дозволяє моделі зосереджуватися на важливих частинах тексту. Завдяки тому, що BERT навчається на великих кількостях текстів, BERT можна використовувати для отримання контекстуальних представлень речень або текстів. Існує кілька підходів до використання BERT для отримання таких представлень:

1) Кодування останнього шару: найпростіший спосіб полягає в використанні контекстуальних представлень останнього шару BERT для кожного токена

в реченні або тексті. Ці представлення містять інформацію про контекст та залежності між словами. Зазвичай, вектор [CLS]-токена використовується як представлення всього речення або тексту. Під час навчання моделі [CLS]-токен вводиться на початку вхідної послідовності та використовується для передбачення чи наступне речення є логічним продовженням поточного.

2) Пулінг представлень: інший підхід полягає в застосуванні пулінгу (наприклад, макс-пулінг (max polling): максимальне значення з кожної компоненти вектора або середнього значення (mean polling): середнє значення векторів) до контекстуальних представлень всіх токенів у реченні або тексті. Це дозволяє отримати фіксований розмір представлення, який резюмує контекст усього речення або тексту.

3) Фінальне налаштування: ще один підхід полягає в доопрацюванні попередньо навченої моделі BERT для конкретної задачі. Це означає, що після навчання BERT можна навчити його на власних даних для покращення результатів в конкретній задачі, наприклад, класифікації тексту чи витягування інформації.

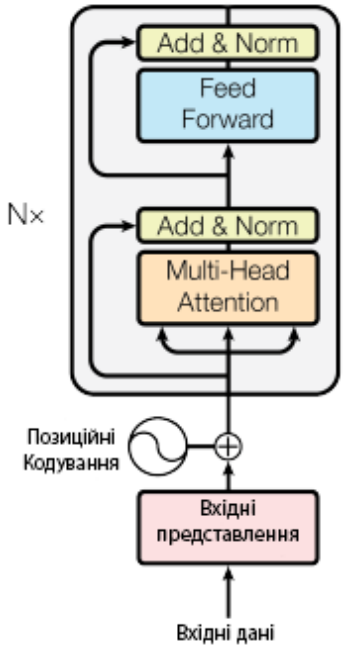

Рис. 1.4. Енкодер з трансформерної архітектури [1]

Використання BERT для отримання представлень речень або текстів надає можливість використовувати ці представлення в різних задачах обробки мови, таких як класифікація, кластеризація, генерація тексту тощо. Враховуючи контекстуальні залежності та багатошарову структуру BERT, отримані представлення мають високу якість та здатні покращити результати у багатьох задачах обробки природної мови. BERT, як базову модель, використовує енкодери з трансформерної архітектури, запропоновану у статті [1] (Рис. 1.4).

Іншою складовою архітектури для навчання КМП - є модель AST (Audio Spectrogram Transformer) [4] - це модель, яка використовує трансформерну архітектуру для обробки звукових спектрограм. Звукові спектрограми - це візуальне представлення звукових сигналів у вигляді двовимірних матриць, де по горизонтальній вісі відображається час, а по вертикальній - частота звуку.

AST використовує ідею Трансформера, що спочатку була запропонована для обробки послідовностей в обробці природної мови, і адаптує її для обробки звукових спектрограм. AST має ту саму архітектуру як і Vision Transformer (ViT) [5]. Основними компонентами AST є кодувальний блок трансформера і механізм уваги.

У кодувальному блоку трансформера звукова спектрограма подається в якості вхідного сигналу, а кожен піксель (або елемент матриці спектрограми) розглядається як токен. Потім модель застосовує механізм уваги для взаємодії між різними токенами, зокрема для моделювання взаємозалежностей у часі та частоті (Рис. 1.5). AST дозволяє моделі виявляти важливі звукові патерни, залежності та характеристики у звукових спектрограмах, що може бути корисним для багатьох аудіо-задач, таких як розпізнавання мови, класифікація звуків, музичний синтез тощо. Використання AST дозволяє отримати контекстуальні представлення звукових спектрограм, що можуть покращити точність та ефективність в аудіо-задачах.

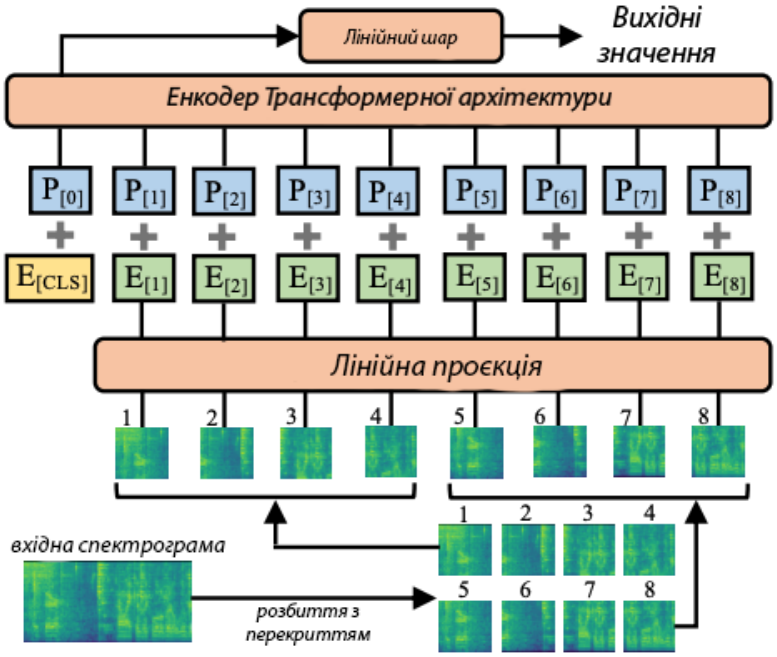

Рис. 1.5. Архітектура AST [4]

1.3. Архітектура моделі для навчання крос-модальних представлень

Архітектура моделі (Рис. 1.6) об'єднує обробку мовлення та обробку музики для здійснення крос-модального розуміння тексту та музики. Основна ідея моделі полягає у використанні зв'язку між музикою та текстом у вигляді «контрастних пар». Модель навчається знаходити відповідність між музикою та текстовими описами. Архітектура моделі базується на двох основних компонентах: моделі обробки мовлення та моделі обробки музики.

Модель обробки мовлення: ця частина моделі використовується для розуміння текстових описів. Вона основана на архітектурі BERT [3] Модель навчається представляти текст у векторному просторі, в якому семантично близькі текстові описи мають схожі вектори.

Модель обробки музики: ця частина моделі використовується для аналізу та розуміння музики. Вона основана на моделі AST [4]. Модель розбиває аудіо

спектрограми на частини та перетворює їх в векторну форму. Потім вона навчається представляти аудіо у векторному просторі, де подібні аудіо мають близькі вектори.

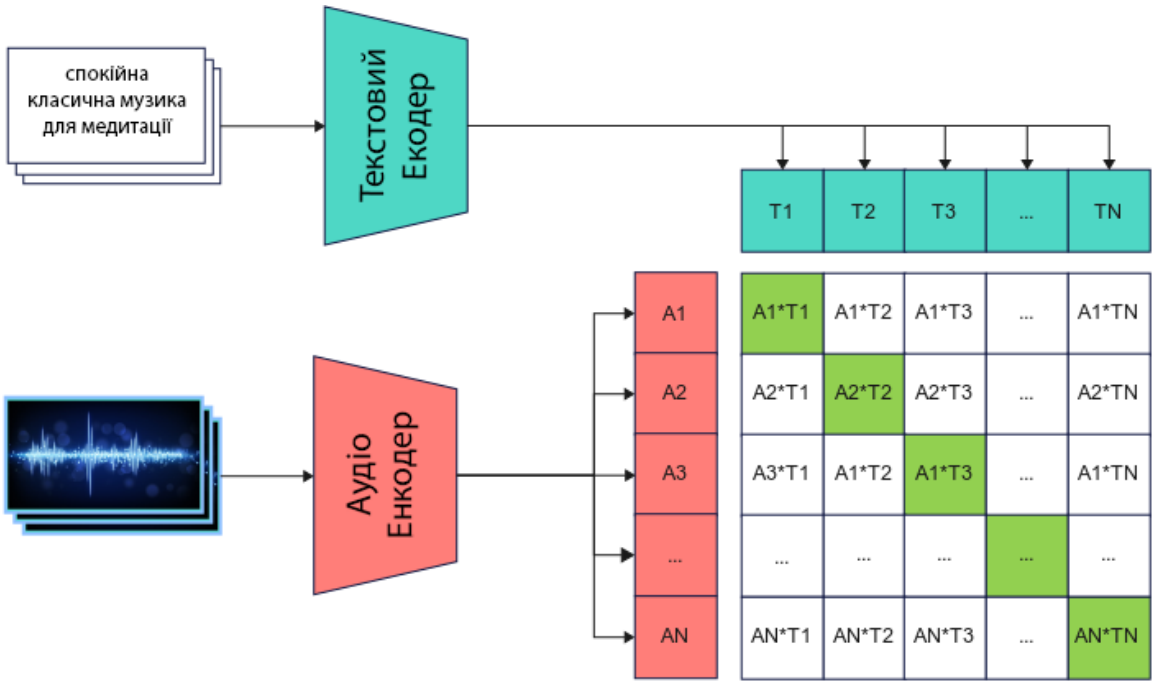

Рис. 1.6. Архітектура моделі для навчання крос-модельних представлень музики та тексту

Обидві частини моделі (обробка мовлення та обробка музики) навчаються разом на великому наборі даних, що містить контрастні пари музики та текстових описів. Під час навчання модель стає здатною знаходити відповідність між музикою та текстом, що дозволяє використовувати її для різних завдань, таких як пошук музики за текстовим описом або класифікація музики за допомогою текстових міток. Архітектура моделі є потужним інструментом для здійснення крос-модального розуміння тексту та аудіо, що відкриває широкі можливості в області обробки природної мови та аудіо даних.

Функція втрат (loss function) моделі для навчання КМП є важливою складовою. Основна мета функції втрат полягає в забезпеченні того, щоб модель навчалась встановлювати відповідність між текстовими описами та музикою. На (див. Рис. 1.6) показана концепція контрастивної функції втрат. На вхід моделі подається пакет аудіо файлів з відповідним текстовим описом до

нього, після цього аудіо енкодер та текстовий енкодер отримають на вхід аудіо та текстовий опис відповідно та повертають векторні представлення аудіо та тексту одного розміру. Після цього між кожною парою аудіо та тексту в пакеті обчислюється косинусова міра (cosine similarity) та записується в матрицю. Далі за допомогою формули 1.1 рахується значення функції втрат.

$$\sum_{i=0}^{N} -\log\left\{\frac{h[\mathbf{f}(\mathbf{t^{(i)}}), \mathbf{g}(\mathbf{a^{(i)}})]}{\sum_{i\neq j} h[\mathbf{f}(\mathbf{t^{(i)}}), \mathbf{g}(\mathbf{a^{(j)}})] + h[\mathbf{f}(\mathbf{t^{(j)}})]}\right\} \tag{1.1}$$

де $\mathbf{f}(\mathbf{t^{(i)}})$ - нормоване векторне представлення тексту $t^{(i)}$,
$\mathbf{g}(\mathbf{a^{(i)}})$ - нормоване векторне представлення музики $a^{(i)}$,
$h[\mathbf{a}, \mathbf{b}] = \exp\left(\mathbf{a}^T\mathbf{b}/\tau\right)$

За допомогою методів оптимізації під час навчання параметри моделі змінюються так щоб мінімізувати дану функцію втрат. Після навчання параметрів текстовий та аудіо енкодер будуть видавати близькі за косинусовою мірою вектори для музики та тексту який описує аудіо. Реалізацію функції втрат для моєї моделі було взято з [6].

Одним з способів використання крос-модальних представлень є пошук даних однієї модальності за допомогою даних іншої модальності (Рис. 1.7). Наприклад пошук музики за текстовим описом, як в нашому випадку. Для знаходження музики за текстовим описом нам спочатку необхідно зібрати бібліотеку музики та для кожного аудіо файлу отримати векторне представлення за допомогою аудіо енкодера. Далі взяти текстовий опис і отримати його векторне представлення за допомогою текстового енкодера і порахувати косинусову міру між векторним представленням тексту та векторним представленням кожного аудіо файлу, музика яке має найбільшу схожість до текстового опису буде мати велике значення косинусової міри, музика яке не підходить під текстовий опис, відповідно, буде мати мале значення косинусової міри. Завдяки цьому можна повернути задану кількість рекомендацій аудіо відсортованих за косинусовою мірою.

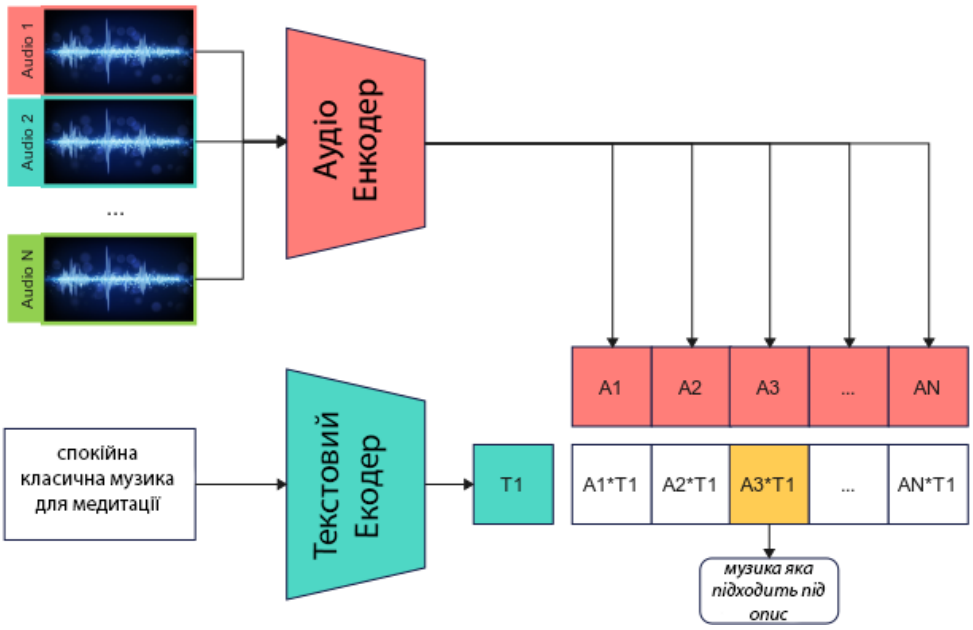

Рис. 1.7. Пошук музики за текстовим описом використовуючи КМП

Іншим способом використання КМП є класифікація музики (Рис. 1.8). Механізм подібний до задачі пошуку музики. За допомогою текстового енкодера шукаються представлення для кожної текстової мітки. Далі за допомогою аудіо енкодера отримується представлення музики у векторній формі, потім для між представленням аудіо та кожним представленням мітки шукається косинусова міра. Тег з яким косинусова міра має найбільше значення і є клас до якого належить музика.

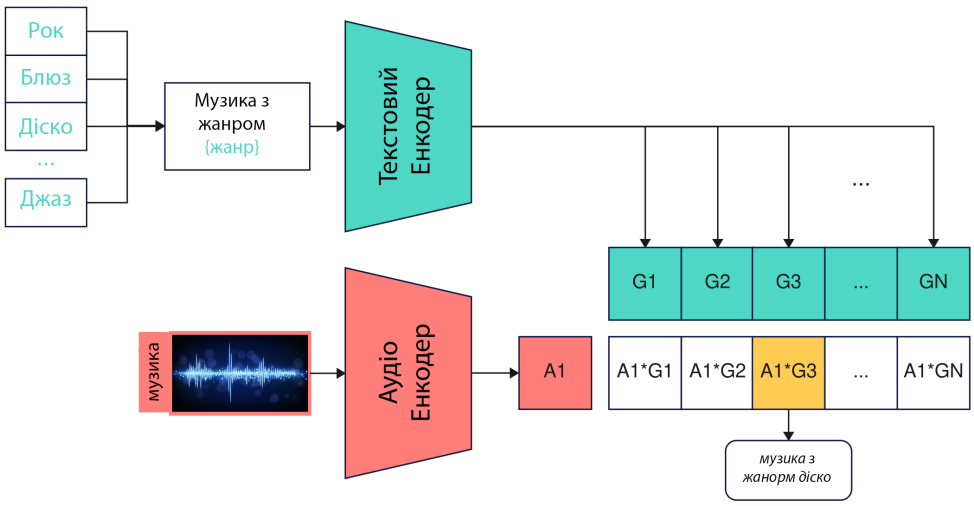

Рис. 1.8. Класифікація жанрів музики використовуючи КМП

Для оцінки якості моделі на задачі пошуку музики було використано такі метрики: Mean Rank; Median Rank; Recall at K; Mean Average Precision at 10; Ці метрики використовуються для оцінки на скільки гарно модель підбирає музику за текстовим описом. У всіх метриках використовується Rank. **Rank** – це позиція істинного аудіо файлу в списку рекомендацій отриманих за допомогою косинусової міри між представленням текстового опису та представленням усіх аудіо в тестовому датасеті.

Mean Rank (MeanR) – вибіркове середнє Rank по всьому тестовому датасету рахується за формулою: $\frac{1}{N}\sum_{i=0}^{N} Rank_i$

Median Rank (MedR) – медіана Rank по всьому тестовому датасету:

$$\begin{cases} \underline{\mathbf{RANK}}[\frac{N+1}{2}], \text{ якщо N парне} \\ \frac{\underline{\mathbf{RANK}}[\frac{N}{2}]+\underline{\mathbf{RANK}}[\frac{N}{2}+1]}{2}, \text{ якщо N непране} \end{cases} \tag{1.2}$$

Де $\underline{\mathbf{RANK}}$ - відсортований Rank, N - розмір тестового датасету.

Recall at K (R@K) – рахується за формулою: $\frac{|\{Rank_i : Rank_i < K\}|}{N}$, де N - розмір тестового датасету.

Mean Average Precision at 10 (MAP10) – рахується за формулою 1.3:

$$\frac{1}{N}\sum_{i=0}^{N}\frac{1}{Rank_i+1} * \mathbb{1}\{Rank_i < 10\} \tag{1.3}$$

1.4. Висновки до розділу 1

В розділі були розглянуті ключові аспекти і концепції, пов'язані з роботою системи пошуку музики за описом з використанням крос-модальних представлень. Були описанні основні принципи машинного навчання та задачі які можна вирішувати за його допомоги, були надано загальну інформацію про

нейронні мережі та представлені різні архітектури нейронних мереж, також було надано інформацію про функції активації на про функції втрат.

Також в розділі була представлена архітектура Трансформера, що є основним каркасом для побудови моделі. Використання механізму уваги дозволяє моделі зосередитись на важливих елементах інформації та враховувати довільні залежності в текстових та аудіо даних. Також в цьому підрозділі було розглянуто модель BERT, яка здатна ефективно кодувати текстову інформацію. Також в підрозділі було представлено AST (Audio Spectrogram Transformer) - модель, що спеціалізується на обробці аудіо даних з використанням Трансформера.

В останньому підрозділі було розглянуто архітектуру моделі для навчання крос-модальних представлень, що дозволяє поєднувати інформацію з текстових та аудіо даних для отримання спільного представлення. Також в цьому підрозділі було розглянуто функцію втрат для тренування розглянутої моделі. Ще було описано методику пошуку музики за допомогою крос-модальних представлень, розглянуто використання навчених представлень для класифікації музики. Нарешті, було представлено метрики для оцінки моделі на задачі пошуку музики за описом, що дозволяють визначити ефективність та точність системи.

В цьому розділі було досліджено та описано ключові теоретичні аспекти, необхідні для розуміння і реалізації системи пошуку музики за описом з використанням крос-модальних представлень. Викладені концепції утворюють основу для подальшого розроблення та вдосконалення таких систем.

РОЗДІЛ 2.

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ НА ЗАДАЧІ ПОШУКУ МУЗИКИ РІЗНИХ АРХІТЕКТУР МОДЕЛЕЙ

2.1. Архітектура моделі

Основними компонентами моделі є текстовий та аудіо енкодер. В реалізації моделі в якості текстового енкодеру був взятий Bidirectional Encoder Representations from Transformers (BERT) [3] претренований на задачі передбачення замаскованих слів (MLM) та передбаченню чи наступне речення є логічним продовженням поточного (NSP). Та в якості аудіо енкодеру був взятий Audio Spectrogram Transformer (AST) [4] претренований на класифікації аудіо на датасеті AudioSet. Перевагою використання претренованих моделей є те, що претреновані AST та BERT вже можуть робити «гарні» представлення аудіо та тексту. Завдяки цьому можна дотренувати тільки останні шари моделей. Також можна використовувати менший датасет.

2.2. Датасет

YouTube id	**Текстовий опис**	**Список аспектів**
3Kv4fdm7Uk	someone is playing a high pitched melody on a steel drum. The file is of poor audio-quality.	['steeldrum', 'higher register', 'amateur recording']
Dtir74TiUM	This is a drum & bass music piece. There is a constantly repeating hollering sample accompanied by r...	['drum & bass', 'electronic dance music', 'hollering sample', 'scratching', 'synth', 'electronic dru...

Таблиця 2.1

Приклад даних в MusicCaps

Для тренування моделі необхідний датасет який має аудіо та відповідний текстовий опис. В якості датасету був взятий MusicCaps, вперше представлений в [7] який має 5,474 посилань на аудіо та відповідних текстових описів для них (Табл. 2.1).

Для тренування та тестування моделі датасет був розбитий на частини з відповідною кількістю входжень музики та описів:
тренувальний датасет: 3,794
валідаційний датасет: 500
тестовий датасет: 1,000

2.3. Тренування різних архітектур моделей та результати тестів для задачі пошуку музики

Для тренування та тестування всіх моделей використовувався одне й те саме розбиття датасету. Тренування проходило з такими налаштуваннями:
швидкість навчання (learning rate) = 3×10^{-4}
максимальна кількість епох = 80
навчання зупинялось якщо функція втрат на валідаційному датасету перестала зменшуватись

Першим варіантом архітектури моделі є модель яка мала дані складові:
Аудіо енкодер складається з претренованого AST та одного лінійного шару проекції який зменшує розмір вектору представлення з 768 до 128
Текстовий енкодер складається з претренованого BERT та одного лінійного шару проекції який зменшує розмір вектору представлення з 768 до 128

Використовуючи різні значення гіперпараметрів після тренування та тестування були отримані такі результати (Табл. 2.2):

	R@1	**R@5**	**R@10**	**MeanR**	**MedR**	**maP10**
Model 1	0.137	0.434	0.591	31.875	6	0.259
Model 2	0.159	0.428	0.577	24.935	6	0.273
Model 3	0.089	0.300	0.467	29.314	10	0.181
Model 4	0.069	0.261	0.413	80.735	15	0.152
Model 5	0.049	0.174	0.303	119.496	23	0.108

Таблиця 2.2

Результати тестів першої варіації моделі

з такими значеннями гіперпараметрів (Табл. 2.3):

	TAU	**Кількість р.ш. BERT**	**Кількість р.ш AST**
Model 1	1.0	5	0
Model 2	2.0	5	0
Model 3	0.2	5	0
Model 4	0.2	3	3
Model 5	0.2	5	5

Таблиця 2.3

Гіперараметри першого варіанту моделі

Кількість р.ш. BERT/AST (Кількість розморожених шарів) – це кількість шарів BERT/AST значення яких змінювались під час тренування моделі. Другим варіантом архітектури стала модель з такими складовими:
Аудіо енкодер складається з претренованого AST з додаванням нових шарів уваги (attention layer) та одного лінійного шару проекції який зменшує розмір вектору представлення з 768 до 128
Текстовий енкодер складається з претренованого BERT з додаванням нових шарів уваги та одного лінійного шару проекції який зменшує розмір вектору представлення з 768 до 128

Використовуючи різні значення гіперпараметрів після тренування та тестування були отримані такі результати (Табл. 2.4):

	R@1	**R@5**	**R@10**	**MeanR**	**MedR**	**maP10**
Model 6	0.178	0.482	0.627	35.138	5	0.303
Model 7	0.154	0.455	0.609	53.423	6	0.281
Model 8	0.166	0.436	0.579	44.134	6	0.278
Model 9	0.153	0.424	0.574	44.251	7	0.269
Model 10	0.111	0.376	0.541	34.183	8	0.227

Таблиця 2.4

Результати тестів другого варіанту моделі

з такими значеннями гіперпараметрів (табл. 2.5):

	TAU	**Кіль-кість р.ш. BERT**	**Кіль-кість р.ш AST**	**Кіль-кість д.ш BERT**	**Кіль-кість д.ш AST**
Model 6	1	5	0	0	1
Model 7	1	5	0	0	3
Model 8	1	5	0	1	1
Model 9	1	3	0	3	3
Model 10	1	5	0	3	0

Таблиця 2.5

Гіперараметри другої варіації моделі

Кількість д.ш BERT/AST – це кількість додаткових шарів уваги доданих до претренованого BERT/AST

Окрім даних архітектур були і інші які давали погані результати. Однією такою моделлю була модель яка мала такі складові:

Аудіо енкодер складається з претренованого AST та повнозв'язною нейронною мережею яка трансформувала вихід енкодера.

Текстовий енкодер складається з претренованого BERT та повнозв'язною нейронною мережею яка трансформувала вихід енкодера.

Тестувалась різна кількість розморожених шарів енкодерів але не одна із цих модифікацій не дала прийнятний результат, порівняно з моделями представленими вище.

Іншою модифікацією була модель яка використовувала замість звичайної повнозв'язної мережі, мережу в якій були пропускні з'єднання (skip connections), які передають інформацію напряму з одного шару мережі до іншого, оминуваючи деякі проміжні шари. Дана модель теж тестувалась з різними значеннями гіперпараметрів але результати тестів були теж порівняно поганими.

Причиною поганих результатів даних моделей може бути те, що енкодер трансформерної архітектури на вихід повертає вектор сталої довжини, тоді як в самому енкодері обробляється вся вхідна послідовність.

2.4. Висновки до розділу 2

Розділ надав важливу інформацію щодо реалізації системи пошуку музики за описом і результатів тестування різних архітектур моделей.

У першому підрозділі була представлена архітектура моделі, яка служить основою для подальшого розроблення та вдосконалення системи. Ця архітектура використовує крос-модальні представлення для зв'язку текстової і аудіо інформації.

В наступному підрозділі було представлено датасет для навчання, який містить вхідні тексти та аудіофайли для тренування і тестування моделей. Цей датасет є важливим компонентом для ефективного навчання та оцінки моделей.

Далі, в останньому підрозділі було надано інформацію про різні архітектури моделей і оцінено їх ефективність на задачі пошуку музики за описом. Результати тестування надали уявлення про те, які моделі показують найкращі результати та їхню здатність вирішувати поставлену задачу. З результатів

тестів стало відомо, що кращий результат мала модель з п'ятьма розмороженими шарами текстового енкодера та з повністю замороженим енкодером, також модель з додаванням додаткового шару уваги до аудіо енкодера дала найкращий результат.

Цей розділ надав важливу інформацію про реалізацію та тестування системи пошуку музики за описом з використанням різних архітектур моделей. Отримані результати є важливим кроком у вдосконаленні системи і відкривають можливості для подальших досліджень та вдосконалень в цій області.

РОЗДІЛ 3.

ТЕСТУВАННЯ РОЗРОБЛЕНИХ МОДЕЛЕЙ НА ІНШИХ ЗАДАЧАХ. ПОРІВНЯННЯ РОЗРОБЛЕНИХ МОДЕЛЕЙ З МОДЕЛЯМИ ВІД СТОРОННІХ РОЗРОБНИКІВ

3.1. Датасети для тестування

Натреновані моделі для отримання КМП можна використовувати для інших задач, таких як: Бінарна класифікація (Binary classification); Багатокласова класифікація (Multiclass classification); Класифікація з декількома мітками (Multilabel classification);

Для тестування отриманих моделей та моделей у відкритому доступі були використані GTZAN Dataset [8] та MagnaTagATune Dataset [9]

GTZAN складається з 1,000 аудіофайлів музичних треків, які належать до 10 різних жанрів: блус, класика, кантрі, джаз, метал, поп, реггі, рок, хіп-хоп та електронна музика. Кожен жанр представлений 100 треками. Тривалість треків становить близько 30 секунд.

MagnaTagATune містить близько 25,000 аудіофайлів треків з різних музичних жанрів та стилів. Кожен трек супроводжується набором тегів, які описують його властивості та характеристики, такі як жанр, інструменти, настрій, енергія та інші атрибути. Датасет MagnaTagATune [9] був створений шляхом залучення спільноти користувачів, які анотували треки, надаючи їм теги на основі своїх спостережень та вражень від прослуховування музики. Це дозволяє використовувати датасет для різноманітних завдань, таких як класифікація музичних жанрів, рекомендації пісень, знаходження схожих треків та багато іншого.

GTZAN був використаний для тестування моделі на задачі багатокласової класифікації.
MagnaTagATune для задачі класифікації з декількома мітками.

3.2. Моделі для порівняння результатів

Першою моделю для порівняння є модель **CLAP** [10] Для даної моделі я провів ті самі тести як і для розробленої мною. Для цієї моделі у відкритому доступі існує натреновані ваги:
Для загального аудіо (*630k-audioset-best.pt*): ваги натреновані на 128,010 парах (аудіо, текст). 36,796 пар з FSD50k, 29,646 пар з ClothoV2, 44,292 з AudioCaps, 17,276 пар з MACS.
Для музики (*music_ audioset_ epoch_ 15_ esc_ 90.14.pt*): натреновано на музиці + Audioset + LAION-Audio-630k.

Також для порівняння результатів була взята модель **MuLan** [11]. Яка була натренована на 44 мільйонів аудіозаписів (370 тис. годин) та слабо пов'язаних вільно сформульованих текстових анотацій. Для даної моделі немає вагів у відкритому доступі але в публікації приведені результати тестів на датасеті MagnaTagATune.

3.3. Тестування розробленої моделі на GTZAN та MagnaTagATune

Відповідно до підрозділу 1.3 була протестована моя натренована модель. В якості метрити було використано F1-socre. Для тестування використовувались різні варіанти подачі жанрів в текстовий енкодер. Всі тестування проводились на моделі з такими параметрами (Табл. 3.1):

TAU	**Кіль-кість р.ш. BERT**	**Кількість р.ш AST**	**Кількість д.ш BERT**	**Кількість д.ш AST**
1.2	5	0	0	0

Таблиця 3.1

Параметри розробленої моделі

Дана модель була натренована на MusicCaps з таким розбитям:
тренувальний датасет: 4794
валідаційний датасет: 500
Так як GTZAN мав аудіо довжини 30 сек., то кожне аудіо було розбито по 10 сек. для цих шматків було знайдене представлення та взято середній вектор з представлень розбиття.
Після тестування були отримані такі результати (Табл. 3.2):

Genre prompt	**F1-score**
'The {genre} song'	0.679
'this is a music of {genre}'	0.372
'this is a sound of {genre}'	0.315
'{genre}'	0.360

Таблиця 3.2

Результати тестів розробленої моделі на GTZAN

Також було проведено тестування на датасеті MagnaTagATune Для тестування була взята та сама модель (див. Табл. 3.1) В якості метрити було використано ROC-AUC score. Для тестування використовувались різні варіанти подачі тегів в текстовий енкодер. Так як MagnaTagATune мав аудіо довжини 29 сек., то кожне аудіо було розбито по 10 сек. для цих шматків було знайдене представлення та взято середній вектор з представлень розбиття.

Після тестування були отримані такі результати (Табл. 3.3):

Tag prompt	**ROC-AUC score**
'{tag}'	0.680
'song with this feature: {tag}'	0.709
'The {tag} song'	0.756

Таблиця 3.3

Результати тестів розробленої моделі на MagnaTagATune

3.4. Тестування моделі CLAP на MusicCaps, GTZAN та MagnaTagATune

Позначення моделей з відповідними натренованими вагами:

CLAP Music - ваги: *music_audioset_epoch_15_esc_90.14.pt*

CLAP Default - ваги: *630k-audioset-best.pt*

Було проведено тестування моделі на датасеті MusicCaps. Для тестування було взяте те саме розбиття датасету, що і для тестування моєї моделі. Були отримані такі результати (Табл. 3.4):

Model	**R@1**	**R@5**	**R@10**	**MeanR**	**MedR**	**maP10**
CLAP Music	0.124	0.361	0.482	49.090	11	0.225
CLAP Default	0.120	0.301	0.449	43.841	12	0.202

Таблиця 3.4

Результати тестів CLAP і на MusicCaps

Для CLAP були проведені ті самі тести на GTZAN та отримані такі результати (Табл. 3.5):

Model	**Genre prompt**	**F1-score**
CLAP Music	'The {genre} song'	0.659
CLAP Music	'this is a music of {genre}'	0.634
CLAP Music	'this is a sound of {genre}'	0.637
CLAP Music	'{genre}'	0.520
CLAP Default	'The {genre} song'	0.460
CLAP Default	'this is a music of {genre}'	0.629
CLAP Default	'this is a sound of {genre}'	0.357
CLAP Default	'{genre}'	0.334

Таблиця 3.5

Результати тестів CLAP і на GTZAN

Для CLAP були проведені ті самі тести на MagnaTagATune та отримані такі результати (Табл. 3.6):

Model	**Genre prompt**	**ROC-AUC score**
CLAP Music	'{tag}'	0.767
CLAP Music	'song with this feature: {tag}'	0.642
CLAP Music	'The {tag} song'	0.798
CLAP Default	'{tag}'	0.749
CLAP Default	'song with this feature: {tag}'	0.714
CLAP Default	'The {tag} song'	0.767

Таблиця 3.6

Результати тестів CLAP і на MagnaTagATune

3.5. Результати тестів моделі MuLan на MagnaTagATune

В публікації [11] описані такі результати тестів моделі на MagnaTagATune (Табл. 3.7):

Model	**ROC-AUC score**
M-AST	0.778
M-Resnet-50	0.782

Таблиця 3.7

Результати тестів MuLan і на MagnaTagATune

позначення для моделей:

M-AST - AST audio encoder

M-Resnet-50 Resnet-50 audio encoder

У обох випадках було використано архітектуру BERT-base-uncased в якості кодувальника тексту. Навченні всі моделі протягом 14 епох на колекції аудіо-текстових пар, взятих з 44 млн. музичних записів і оброблених текстових міток у всіх категоріях: AudioSet (ASET), короткі теги (SF), довгі речення (LF), інформація про плейлисти (PL). Було використано оптимізатор Adam зі схемою пониження швидкості навчання з кроком зниження 0.9, застосованим кожні 40 тис. кроків, та початковими значеннями 5×10^{-5} для M-Resnet-50 та 4×10^{-5} для M-AST.

3.6. Висновки до розділу 3

Порівняння отриманих моделей з моделями від сторонніх розробників надало важливу інформацію щодо ефективності розробленої моделі.

У першому підрозділі були представлені використані датасети для тестування, зокрема GTZAN та MagnaTagATune. Ці датасети мають важливе значення для порівняння результатів та оцінки ефективності моделей.

У наступних підрозділах було розглянуто моделі від сторонніх розробників, які використовувались для порівняння результатів. Дані моделі були протестовані на датасетах MusicCaps, GTZAN та MagnaTagATune. З отриманих

результатів видно, що розроблена модель показала кращий результат за модель CLAP в задачі пошуку музики за текстовим описом на датасеті MusicCaps, та в задачі класифікації жанрів на датасеті GTZAN. Також дала схожий до MuLan результат на датасеті MagnaTagATune. Так як, MagnaTagATune містить теги, які не відносяться до музики, а до аудіо в загальному, тому отриману модель доцільно використовувати саме для задач пов'язаних з музикою.

ВИСНОВКИ

У даній роботі був проведений детальний аналіз та дослідження в області пошуку музики за допомогою крос-модальних представлень.

В першому розділі були розглянуті теоретичні відомості пов'язані з машинним навчанням та з нейронними мережами, також було більш детально розглянуто про архітектуру Трансформер та про моделі, які є складовою частиною розробленої моделі BERT (Bidirectional Encoder Representations from Transformers) та AST (Audio Spectrogram Transformer), основані на архітектурі Трансформер. Було розглянуто архітектуру моделі для навчання крос-модальних представлень і задачі які можна вирішувати за допомогою отриманих крос-модальних представлень. Також були розглянуті функція втрат і метрики для оцінки моделей на задачах пошуку музики за текстовим описом. Даний розділ надає важливу інформацію для подальшого покращення розробленої моделі.

У другому розділі було описано реалізацію моделі та її модифікацій для навчання крос-модальних представлень, також було описано тренування різних архітектур моделей на спеціально підготовленому датасеті. Ще були представлені результати тестування моделей на задачі пошуку музики. Результати тестів показали, що додавання додаткових шарів уваги до претренованих енкодерів дають покращення в результатах тестів. Також з результатів можна побачити, що найкращим варіантом буде залишити AST повністю замороженим та розморозити 5 шарів в BERT.

У третьому розділі було проведено тестування отриманих моделей на інших задачах і порівняння їх з моделями, що доступні у відкритому доступі. Для цього використовувалися різні датасети та моделі для порівняння результатів. Тестування було проведено на GTZAN, MagnaTagATune і MusicCaps. Результати тестів підтвердили ефективність розроблених моделей у вирішенні задач пошуку музики за текстовим описом та задач класифікації. Спираю-

чись на результати тестів, можна стверджувати, що хоча розроблені моделі тренувались на порівняно невеликому датасеті (5,474 пар музики та описів) в порівнянні з (128,010 пар) в моделі CLAP [10] та (44 млн. пар) в моделі MuLan [11], розроблена модель показала кращий за модель CLAP результат в задачі класифікації жанрів та задачі пошуку музики за текстовим описом на датасеті MusicCaps. Також отримана модель дала схожий результат до моделі MuLan у задачі класифікації на датасеті MagnaTagATune. Отримані результати тестування демонструють високу точність і здатність розробленої моделі до класифікації жанрів та пошуку музичних композицій за описом.

ПЕРЕЛІК ПОСИЛАНЬ

1. Attention is all you need / A. Vaswani [та ін.] // Advances in neural information processing systems. — 2017. — Т. 30.
2. Self Attention in Convolutional Neural Networks. — `https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00`.
3. Bert: Pre-training of deep bidirectional transformers for language understanding / J. Devlin [та ін.] // arXiv preprint arXiv:1810.04805. — 2018.
4. Gong Y., Chung Y.-A., Glass J. Ast: Audio spectrogram transformer // arXiv preprint arXiv:2104.01778. — 2021.
5. An image is worth 16x16 words: Transformers for image recognition at scale / A. Dosovitskiy [та ін.] // arXiv preprint arXiv:2010.11929. — 2020.
6. Learning transferable visual models from natural language supervision / A. Radford [та ін.] // International conference on machine learning. — PMLR. 2021. — С. 8748—8763.
7. Musiclm: Generating music from text / A. Agostinelli [та ін.] // arXiv preprint arXiv:2301.11325. — 2023.
8. Sturm B. L. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use // arXiv preprint arXiv:1306.1461. — 2013.
9. Evaluation of algorithms using games: The case of music tagging. / E. Law [та ін.] // ISMIR. — Citeseer. 2009. — С. 387—392.
10. Clap learning audio concepts from natural language supervision / B. Elizalde [та ін.] // ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2023. — С. 1—5.
11. Mulan: A joint embedding of music audio and natural language / Q. Huang [та ін.] // arXiv preprint arXiv:2208.12415. — 2022.