КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (19), 2023
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2023.19.146164
УДК 004.896

Іосіфов Євген Анатолійович
аспірант кафедри інформаційної та кібернетичної безпеки ім. проф. Володимира Бурячка
Київський університет імені Бориса Грінченка, Київ, Україна
ORCID ID: 0000-0001-6203-9945
y.iosifov.asp@kubg.edu.ua

КОМПЛЕКСНИЙ МЕТОД ПО АВТОМАТИЧНОМУ РОЗПІЗНАВАННЮ ПРИРОДНЬОЇ МОВИ ТА ЕМОЦІЙНОГО СТАНУ

Анотація. Поточні тенденції в NLP наголошують на універсальних моделях та навчанні з передварительно навчених моделей. У цій статті досліджуються ці тенденції та передові моделі попереднього навчання. Вхідні дані перетворюються на слова або контекстуальні вбудовування, які слугують вхідними даними для енкодерів та декодерів. В якості об'єкту дослідження використовується корпус публікацій автора статті за останні шість років. Основними методами дослідження є аналіз наукової літератури, прототипування і експериментальне використання систем за напрямком досліджень. Гравці розпізнавання мови розділилися на гравців з величезними обчислювальними ресурсами для котрих тренування на великих нелейбованих даних є звичною процедурою, і гравців які сфокусовані на тренуванні малих локальних моделей розпізнавання мови на попередньо розмічених аудіо даних через нестачу ресурсів. Підходи і фреймворки роботи з нелейбованими даними і обмеженими обчислювальними ресурсами майже не представлені, а методики базовані на ітеративних тренуваннях не розвинуті і потребують наукових зусиль для розвитку. Дослідження має на меті розвинути методики ітеративного тренування на нерозмічених аудіо даних для отримання продуктивно готових моделей розпізнавання мови з більшою точністю і обмеженими ресурсами. Окремим блоком запроновані методи підготовки даних для використанні в тренуванні систем розпізнавання мови і конвейер автоматичного тренування систем розпізнавання мови використовуючи псевдо розмітку аудіо даних. Прототип і вирішення реальної бізнес задачі з виявлення емоцій демонструють можливості і обмеження систем розпізнавання сови та емоційних станів. З використанням запропонованих методів псевдо-лейбування вдається без значних інвестицій в обчислювальні ресурси отримати точність розпізнавання близьку до лідерів ринку а для мов з незначною кількістю відкритих даних навіть перевершити.

Ключові слова: автоматичне розпізнавання мови; APM; NLP; рекурентні нейронна мережа; RNN.

ВСТУП

NLP (англ. natural language programming) — це комп'ютерна обробка та розуміння природньої мови з використанням таких випадків, як машинний переклад та розпізнавання мовлення. Метою є зрозуміння машинами мови на людському рівні. У цій статті досліджуються техніки побудови мовних моделей, з фокусом на передових архітектурах.

Поточні тенденції в NLP наголошують на універсальних моделях та навчанні з передварительно навчених моделей. У цій статті досліджуються ці тенденції та передові моделі попереднього навчання. Вхідні дані перетворюються на слова або контекстуальні вбудовування, які слугують вхідними даними для енкодерів та декодерів.

Розпізнавання мови стає все більш актуальним завданням у сфері обробки природної мови (NLP) через поширення голосових асистентів, розумних домів та інших технологій, які вимагають розуміння та аналізу мови. Ефективні системи розпізнавання

мови можуть значно поліпшити якість інтерфейсів та сприяти комунікації між людьми та комп'ютерами.

Постановка проблеми. Гравці розпізнавання мови розділилися на гравців з величезними обчислювальними ресурсами для котрих тренування на великих нелейбованих даних є звичною процедурою, і гравців які сфокусовані на тренуванні малих локальних моделей розпізнавання мови на попередньо розмічених аудіо даних через нестачу ресурсів. Підходи і фреймворки роботи з нелейбованими даними і обмеженими обчислювальними ресурсами майже не представлені, а методики базовані на ітеративних тренуваннях не розвинуті і потребують наукових зусиль для розвитку.

Аналіз останніх досліджень і публікацій.

В роботі [1] розглянутий основний проривний метод на якому базуються всі новітні методи обробки природньої мови – механізм уваги. Він розкрив нові можливості як для текстового так і для аудіо аналізу даних. Однак він має суттєві обмеження і недоліки і потребує подальшого розвитку.

Гібридна методика роботи з аудіо даними для розпізнавання описана в [2]. Але зважаючи на її обмеження в [3] розглядаються порівняння з новітніми методиками основаними на механізмі уваги.

Використання методів тренування на нелейбованих даних і впливу їх на якість показана в [4]. В роботі запропонований новітня методика маскування даних для тренування моделей і адаптації під неочевидні задачі.

Окремо слід зазначити практичний вклад роботи [5] на формування підходу ітеративного псевдо-лейбування. Робота представлена в доволі стиснутому вигляді і потребує подальної наукової роботи, але навіть в такому короткому вигляді дає н еоціненний вклад в тренування моделей на основі псевдо-лейбування.

Дослідження має на **меті** розвинути методики ітеративного тренування на нерозмічених аудіо даних для отримання продуктивно готових моделей розпізнавання мови з більшою точністю і обмеженими ресурсами.

МЕТОДИКА ДОСЛІДЖЕННЯ

В даній статті використовується досвід роботи з методами розпізнавання мови, а також методами обробки природньої мови. В якості **об'єкту** дослідження використовується корпус публікацій автора статті за останні шість років. Основними **методами** дослідження є аналіз наукової літератури, прототипування і експериментальне використання систем за напрямком досліджень.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Контекстуальні вбудовування — це техніка, яка використовується для перетворення слів на числові вектори. Кожне слово представляється як вектор з фіксованою кількістю компонентів, а кожна компонента вектору представляє значення для певної ознаки. Зазвичай, такі вектори використовуються для навчання моделей машинного навчання. Контекстуальні вбудовування — це також розширена версія словарного вбудовування (word embedding), де кожне слово не тільки представляється вектором, але також містить інформацію про контекст, в якому воно вживається. Це дозволяє враховувати більше інформації про слово та його семантику в контексті речення або тексту. Приклади реалізацій контекстуальних вбудовувань це GloVe та Word2Vec [2, 6, 7].

Архітектура енкодер-декодер — це структура нейронної мережі, яка використовується в області обробки природної мови та машинного перекладу [8]. Енкодер отримує на вхід послідовність даних та перетворює її на складний вектор фіксованої довжини, який містить у собі семантичну інформацію про вхідну послідовність. Декодер отримує цей вектор на вхід та генерує вихідну послідовність. Цей підхід дає можливість моделям машинного перекладу «розуміти» вміст вхідного тексту та генерувати відповідну послідовність у вихідній мові. Також кодер-декодер може використовуватись для багатьох інших завдань, таких як стиснення тексту або генерація відповідей на запитання.

Рекурентні нейронні мережі

Рекурентні нейронні мережі ‘recurrent neural networks’ (RNN) — це тип нейронних мереж, що здатні обробляти послідовності даних, такі як текст або аудіо [9, 10]. Одна з особливостей RNN полягає в тому, що вона зберігає попередні стани в процесі обробки послідовності, що дозволяє їй зосереджуватися на попередніх елементах послідовності при обробці наступного елемента. Це робить RNN корисним для завдань, які потребують контекстуального розуміння, наприклад, для машинного перекладу або розпізнавання мови. В RNN також є декілька варіацій, таких як LSTM і GRU [11], які дозволяють вирішувати проблему зниклого градієнту, що може виникати при навчанні мережі на довгих послідовностях.

Концепція уваги

Концепція уваги в NLP — це важливий механізм, що допомагає моделям нейронних мереж краще зосереджуватися на відповідних частинах вхідних даних під час виконання задачі. Він був запропонований для покращення роботи рекурентних нейронних мереж (RNN) в задачах машинного перекладу, але згодом знайшов широке застосування в інших доменах NLP.

Основна ідея механізму уваги полягає в забезпеченні зваженого представлення контексту. Замість того, щоб використовувати одну фіксовану контекстуальну вектор, як це робили традиційні RNN, модель з увагою враховує значущість різних частин вхідного тексту на кожному кроці вивчення або передбачення.

У контексті машинного перекладу механізм уваги дозволяє моделі краще враховувати відповідності між словами вихідної та цільової мови. Він забезпечує зв'язок між певними словами у вихідному тексті і їх перекладами, що поліпшує якість перекладу, особливо для довгих речень.

Механізм уваги також був успішно застосований в інших архітектурах нейронних мереж, таких як Transformer, які використовуються в сучасних моделях NLP, таких як BERT і GPT.

У загальному випадку, механізм уваги допомагає моделям краще впоратися з довготривалими залежностями та забезпечує зв'язок між різними частинами тексту, що дозволяє отримати більш точні результати в широкому спектрі задач NLP.

Концепція самоуваги та трансформеру

Самоувага та трансформер [1] в NLP є ключовими концепціями, які призвели до значного покращення різних завдань обробки природних мов.

Самоувага — це механізм, що дозволяє нейронній мережі зосереджуватися на різних частинах вхідної послідовності для кожного елемента цієї послідовності. Він

враховує взаємозв'язок між словами в рамках одного тексту, замість того, щоб порівнювати їх з іншими текстами. Самоувага забезпечує зважене представлення контексту на основі ступеня відповідності між словами в одному тексті.

Трансформер — це архітектура нейронної мережі, яка використовує механізм Self-Attention для вирішення задач NLP. Відрізняючись від попередніх архітектур, таких як RNN та згорткові нейронні мережі (CNN), трансформер повністю відмовляється від послідовної обробки даних, забезпечуючи паралельність в обробці.

Архітектура трансформеру складається з двох основних компонентів: енкодера та декодера. Енкодер містить декілька однакових шарів, які включають модулі самоуваги та позиційно-залежні згорткові шари. Декодер має аналогічну структуру, але додає ще один модуль уваги, який забезпечує зв'язок між енкодером та декодером.

Трансформер став основою для багатьох сучасних моделей NLP, таких як BERT, GPT-2, GPT-3 та інших, які досягли досі найкращих результатів у ряді задач обробки природних мов, включаючи класифікацію текстів, машинний переклад та інші.

Методи автоматичного розпізнавання мови (АРМ) [12–14] зосереджуються на перетворенні звукового сигналу на письмовий текст. Ці методи можна розділити на три основні категорії [3, 15, 16]: акустичне моделювання, мовне моделювання та декодування.

1. *Акустичне моделювання*. Цей етап включає в себе навчання моделі для перетворення акустичних сигналів (звукових хвиль) на фонеми або інші одиниці мови. Тут використовуються такі техніки, як глибоке навчання, зокрема CNN та RNN, для аналізу звукових характеристик та їх відповідного відображення на мовних одиницях.

2. *Мовне моделювання*. На цьому етапі створюється мовна модель, яка допомагає передбачити ймовірність наступного слова або фрази в контексті попередніх слів [17]. Мовні моделі зазвичай засновані на статистичних або нейронних підходах. У статистичному підході використовуються *n*-грами для визначення ймовірності послідовностей слів, тоді як в нейронному підході використовуються мережі, такі як LSTM, GRU або трансформер, для забезпечення глибшого контексту.

3. *Декодування*. На останньому етапі, алгоритми декодування, такі як Greedy (жадібний) декодер або Beam Search, використовуються для генерації оптимального текстового виводу на основі акустичної та мовної моделі. Ці алгоритми враховують ймовірність послідовностей слів, що генеруються обома моделями, та обирають найкращий варіант.

Часто використовуються глибокі навчальні моделі та передові алгоритми для досягнення високої точності розпізнавання мови в різних сценаріях та доменах. Ось деякі з ключових аспектів сучасних систем розпізнавання мови:

1. *Моделі повного циклу*. Замість того, щоб мати окремі акустичні та мовні моделі, деякі сучасні системи розпізнавання мови використовують наскрізні моделі, такі як Listen, Attend and Spell (LAS) або Deep Speech 2. Ці моделі навчаються безпосередньо від акустичного сигналу до текстового виводу, що дозволяє їм оптимізувати розпізнавання мови без потреби в окремих компонентах [16, 18, 19].

2. *Перенос навчання.* Для покращення ефективності навчання та загальної роботи моделей, передові системи розпізнавання мови використовують попередньо навчені моделі на великих наборах даних та адаптують їх до специфічних завдань або доменів. Це спрощує процес навчання та забезпечує кращі результати.

3. *Мультимовність*. Сучасні системи розпізнавання мови часто підтримують кілька мов, використовуючи загальні архітектури та набори даних для навчання моделей, які можуть працювати з різними мовами. Це забезпечує гнучкість та зручність використання системи в різних культурних та мовних контекстах.

4. *Адаптація* до шуму та акцентів. Сучасні системи розпізнавання мови навчаються адаптуватися до різних акцентів та фонового шуму, щоб забезпечити кращу роботу в різних умовах. Це дозволяє моделям працювати ефективно навіть у складних акустичних умовах та з різними варіаціями мови.

5. Покращення у швидкості та *ефективності*. Сучасні системи розпізнавання мови постійно вдосконалюються для забезпечення більшої швидкості обробки та ефективності. Це включає оптимізацію алгоритмів, використання апаратного прискорення та розробку легших моделей, які можуть працювати на пристроях з обмеженими ресурсами, такими як мобільні телефони або IoT-пристрої.

6. *Розпізнавання* голосових команд та голосових помічників. Розвиток розпізнавання мови стимулює створення технологій, які дозволяють людям взаємодіяти з пристроями та послугами через голосові команди. Голосові помічники, такі як Amazon Alexa, Google Assistant та Apple Siri, використовують системи розпізнавання мови для забезпечення зручного способу взаємодії з користувачами.

7. *Оцінка та порівняння* систем розпізнавання мови. Для оцінки якості розпізнавання мови та порівняння різних систем використовуються стандартні метрики, такі як WER (Word Error Rate) або CER (Character Error Rate). Ці метрики допомагають визначити точність та надійність системи, а також виявити можливі напрямки для покращення.

Всі ці аспекти разом привели до створення потужних та ефективних систем розпізнавання мови, які продовжують розвиватися та вдосконалюватися з підвищенням рівня технічного прогресу та збільшенням доступних даних для навчання моделей.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Підготовка тренувальних даних для сегментації речень

Для переходу до практичної роботи з розпізнаванням мови, основні обмеження полягають в наявності і придатності тренувальних даних.

Тренування сучасних архітектур потребує дуже великої кількості даних, щоб неточності даних могли бути нівельовані кількістю.

Підготовка даних та тренування моделей машинного навчання, особливо на нерозмічених даних, може бути складним процесом. Це обумовлено декількома факторами:

1. *Великі обсяги даних*. Навчання ефективних моделей машинного навчання вимагає великих наборів даних. Однак збір, обробка та зберігання таких великих обсягів інформації можуть бути викликом.

2. *Неповнота та неякісність даних*. Нерозмічені дані можуть бути неповними, пошкодженими або містити шум, що ускладнює їх використання для навчання моделей. Часто потрібно витратити значний час на очищення даних та попередню обробку, перш ніж їх можна використати для тренування.

3. *Відсутність розмітки*. Нерозмічені дані не містять явних позначень або міток, які б дозволили моделі визначити правильні відповіді під час навчання. Це призводить до необхідності використовувати методи навчання без учителя або напівнаглядованого навчання, які можуть бути складнішими та менш точними за навчання з учителем.

4. *Вибір оптимальних параметрів та архітектур*. Вибір оптимальних параметрів моделі та структури нейронної мережі може бути складним, особливо для нерозмічених даних, де може бути відсутня явна зворотній зв'язок про ефективність різних параметрів.

5. *Обчислювальні ресурси та час*. Тренування моделей на великих наборах нерозмічених даних може бути часозатратним процесом, що вимагає значних обчислювальних ресурсів. Це може бути перешкодою для дослідників і розробників, особливо для тих, хто має обмежений доступ до потужних обчислювальних систем.

6. *Оцінка та валідація*. Оцінка моделей, навчених на нерозмічених даних, може бути складнішою, оскільки відсутність явних міток унеможливлює використання стандартних методів оцінки, таких як точність, відгук та F-міра. Дослідники можуть застосовувати альтернативні методи оцінки, такі як зовнішні метрики успіху або кластеризація, але ці методи можуть бути менш надійними та об'єктивними.

7. *Адаптація до нових доменів*. Нерозмічені дані можуть бути зібрані з різних джерел та доменів, що може ускладнити адаптацію моделей до нових доменів. Можливість моделі адаптуватися до нових даних може бути обмеженою, що може призвести до поганих результатів на даних з нових доменів.

8. *Етичні та конфіденційні питання*. Робота з нерозміченими даними може включати етичні та конфіденційні питання, особливо якщо дані містять особисту інформацію або відносяться до вразливих груп. У таких випадках дослідникам і розробникам слід забезпечити відповідні механізми захисту даних та дотримуватися принципів етичного дослідження.

Усе вище зазначене вимагає від дослідників та розробників систематичного підходу до підготовки даних та тренування моделей на нерозмічених даних. Успішне подолання цих викликів може значно покращити якість та ефективність моделей машинного навчання, допомагаючи їм краще розуміти та обробляти природні мови та інші види даних.

Було обрано дві задачі для підготовки даних:

1. Підготовка даних за допомогою розмітки пунктуацією (для використання великої кількості нерозмічених даних).

2. Автоматизований потік для тренування з використання нерозмічених даних.

Підготовка даних є дуже важливим етапом у будь-якому завданні машинного та глибинного навчання. Особливо це стосується обробки природної мови, де вхідні дані повинні бути розділені на речення. На сьогоднішній день велика кількість інформації зберігається у звукових послідовностях (окремо або як додаток до відеопотоку). Багато з таких даних вже мають автоматично згенеровані транскрипції — сирий набор слів без будь-яких знаків пунктуації та розділення на речення. Отже, цей великий масив даних не може бути використаний для вирішення завдань обробки природної мови в поточному стані [20].

Сегментація речень з неформатованого тексту

Як і в більшості завдань глибинного навчання, існує багато можливих підходів для вирішення проблеми розділення сирого тексту на речення. Автори вирішили дослідити проблему двома підходами:

1. Сформулювати проблему як задачу моделювання мови, тобто намагатися передбачити наступний токен (слово або символ) для заданої послідовності введення (початку речення) [4, 21].

2. Сформулювати проблему як задачу позначення послідовності, яка полягає в тому, щоб призначити мітку кожному токену з заданої послідовності введення (у поточній задачі «поточний токен є останнім токеном речення») [22, 23].

Після проведення експериментів для поточної проблеми підхід вирішення задачі як задача моделювання мови показує значно менші результати порівняно з підходом

позначення послідовності. Крім того, цей підхід є значно менш ресурсновитратним, як показано в табл. 1.

Таблиця 1

Порівняння рішення проблеми розділення речень за допомогою різних підходів

Модель	Підхід	Час тренування, с	Час розрахунку, с	Показник F1
DistilBERT [23]	Позначення послідовності	1 309	18	80,40
BERT base [4]	Позначення послідовності	3 121	39	82,10
XLNet [24]	Позначення послідовності	3 504	92	88,35
DistilGPT-2 [25]	Моделювання мови	—	16 270	4,02
GPT-2 [26]	Моделювання мови	—	26 700	4,53
Transformer-XL [27]	Моделювання мови	—	57 000	6,24

Моделі, які використовували підхід позначення послідовності, значно перевершили ті, які використовували підхід моделювання мови. Розглянувши топових виконавців в методі позначення послідовності, показник F1 від 80,40% для DistilBERT до 88,35% для XLNet змінювався майже на 7,95%. Це покращення точності показник F1 на 7,95% відбулося за рахунок збільшення обчислювального часу в п’ять разів (від 18 до 92 секунд) між найшвидшою і найповільнішою моделями.

Менші моделі, які походять від більших, наприклад, DistilBERT (заснований на BERT), показують майже таку саму точність, водночас потребуючи на 58% менше часу для навчання та на 53% менше часу для передбачення порівняно з моделлю BERT base. Це значні показники для виробничого середовища. Таким чином, якщо різниця в точності показника F1 на рівні 2% не є вирішальною, рекомендується вибрати більш легку модель DistilBERT [28].

Оцінка результатів сегментації речень

Найбільший вплив на результати досягається завдяки вибору підходу до вирішення конкретної проблеми (наприклад, сегментації речень). Підхід позначення послідовності значно перевершує підхід моделювання мови як у точності показника F1, так і в часі передбачення для сегментації речень.

Отже, для задач сегментації речень слід використовувати підхід позначення послідовності. Крім того, важливо підготувати і мати на увазі високоякісний набір даних, що стосується кожного конкретного випадку, включаючи незвичайні мітки, які можуть бути корисними для майбутніх застосувань.

В цілому, налаштування попередньо навчених моделей демонструє простоту використання та задовільні результати, що свідчить про те, що завдання підготовки неформатованих даних може бути вирішено з високою точністю.

Підготовка тренувальних даних із нерозмічених аудіофайлів

Вирішення попередньої задачі дає можливість використання наявних аудіоданих, якщо до них є прикріплені тексти, але не дає можливості використовувати нерозмічені дані.

Було розроблено алгоритм програмного забезпечення для АРМ, що дозволяє генерувати навчальні набори даних з неформатованих джерел звуку (спрямовані на конкретну мову або не позначені аудіо).

Основною метою цього автоматизованого конвеєра (пайплайну) є зменшення часу, необхідного для підготовки тренувального набору даних потрібного розміру, та полегшення процесу навчання для новачків, які працюють з існуючими рамками та інструментами АРМ.

Використання однієї мови програмування може сприяти полегшенню процесу навчання. Крім того, оскільки до команди АРМ можуть входити вчені та інженери машинного навчання, корисно використовувати мову програмування, з якою вони вже знайомі. Тому було обрано Python як основну мову для автоматизованого конвеєру АРМ. Наша увага зосереджена на невеликих та середніх наборах даних до кількох тисяч годин. Ми виявили, що обробка невеликого набору даних на одному комп'ютері спрощує використання конвеєра. Ми вважаємо, що це розумна компромісна угода, оскільки нові дослідники та команди мають мало ймовірність мати більше, ніж 10 000 годин аудіо.

Основні компоненти

Ми визначили чотири основні типи компонентів:

- DataProxy. Об’єкт для передачі даних між кроками пайплайну.
- Transformations. Компоненти обробки, які отримують один або кілька DataProxy об'єктів та повертають новий DataProxy.
- Data Input. Використовується для читання даних з зовнішніх джерел.
- Data Output. Використовується для експорту результатів обробки у певному форматі. Комбінація визначених компонентів дозволяє нам визначити пайплайни. Наступним кроком є визначення структур даних та API для досягнення взаємодії між компонентами.

Перед визначенням структур даних для компонентів, необхідно вирішити, як зв'язати окремі компоненти, щоб створити єдиний конвеєр. Як спостерігалось в інших обчислювальних фреймворках [29–31], направлені ациклічні графи є загальним методом побудови складних обчислень з кількох компонентів. Цей підхід також підходить для конвеєрів АРМ, і використання знайомих концепцій може зменшити час навчання. В результаті ми представляємо обчислювальну структуру у вигляді направленого ациклічного графа.

Щоб зробити різні компоненти сумісними між собою, ми повинні визначити формат даних, що передаються між ними. Структура даних включатиме дві основні компоненти: основні дані (наприклад, транскрипції та посилання на аудіофайли) та метадані (наприклад, інформація про кешування). Самі метадані можуть складатися з двох частин. Одна частина — це дані рівня трансформацій (наприклад, коли трансформація почалася / закінчилася). Друга частина метаданих — це дані рівня рядків. Вони зберігатимуть мета-інформацію для кожного аудіофайлу або репліки. Рівні дані необхідні для підтримки часткової інвалідації.

Об’єднання реалізацій трансформацій має наступні переваги:

1. Скорочує час навчання, оскільки користувач вивчає, як викликати кожну трансформацію та як комбінувати кілька трансформацій один раз.

2. Дозволяє мати стандартну реалізацію кешування, яку можна використовувати з різними трансформаціями.

3. Зроблює компоненти більш майбутньоорієнтованими, оскільки ми можемо додати глобальний планувальник завдань для розподілених обчислень без переписування трансформацій з нуля.

Автоматичне кешування та інвалідація дозволяють прискорити експерименти, повторну обробку вихідних даних з додатковими етапами тощо. Кешування включає наступні рішення: які трансформації можуть бути кешовані; як перевірити, чи не змінені дані; як відстежувати, яка частина даних змінилася, щоб здійснювати часткову інвалідацію; як зберігати дані. Нижче ми розглянемо ці рішення.

Беручи до уваги цільову область (робота з власними наборами даних), загальним сценарієм буде додавання або видалення частини даних. У таких випадках кешувальна процедура повинна виявляти змінену частину даних і робити обчислення тільки для зміненої частини. Щоб здійснювати це на всіх етапах конвеєра, нам потрібен спосіб відображення рядка введення даних на рядок або набір рядків виводу на наступній стадії. Розглянемо конвеєр, що складається з наступних етапів:

1. Завантаження даних з *.csv та *.mp3.
2. Нормалізація числових значень.
3. Розбиття довгих сегментів на коротші.
4. Перемішування та розділення на тренувальну та тестову вибірки.
5. Збереження у форматі навчального фреймворку для АРМ.

У даному конвеєрі, видалення одного початкового файлу призведе до видалення одного рядка на кроці 2, видалення одного або декількох рядків на кроці 3, видалення того ж кількості рядків, але в різних позиціях на кроці 4 і різний вихід на кроці 5. Щоб відстежувати рядки протягом кількох перетворень, нам потрібен спосіб ідентифікувати кожен рядок. Оскільки у нас немає унікальних ідентифікаторів, ми пропонуємо використовувати хеш всіх стовпців набору даних як ідентифікатор рядка для перетворення.

Отже, для підтримки недійсності кешу, включаючи часткову недійсність, список даних, буде розширений двома стовпцями: хеш та батьківський хеш. На рівні виходу ми зберігаємо три значення хешу: хеш параметрів init; хеш параметрів процесування; хеш аудіофайлів.

Алгоритм конвеєра

Вхідні дані:

- Лейбовані дані $L = \{x_i, y_i\}_{i=1}^{l}$.
- Частково лейбовані дані S.
- Нелейбовані дані $U = \{x'_j\}_{j=1}^{u}$.

1. Збір доступних відкритих наборів даних АРМ для поточної мови L.

2. Збір доступних відкритих напівмаркованих даних S (аудіо без тексту, наприклад, книги, відео з транскрипціями).

3. Збір доступних відкритих непомічених даних U (аудіо без тексту, наприклад, публічне радіо).

4. Очищення зібраного набору даних S шляхом застосування загальних кроків підготовки наборів даних АРМ, таких як виявлення голосу, ідентифікація дублікатів

фрагментів, класифікація на рівні семестру/музики/шуму та фільтрування за оцінкою середнього рівня якості.

5. Нормалізація цифр та чисел для набору даних *S*.

6. Вирівнювання аудіо до текстових транскрипцій для набору даних *S*. Техніки вирівнювання описані в [32].

Результатом є підготовка акустичної та мовної моделей p_θ на наборі даних *L* та *S*, використовуючи на початку тільки розмічені дані *L* та *S*.

Процес включає наступні кроки:

1. Розпізнати нову нерозмічену порцію даних *U* (об'ємом у 200 годин).

2. Перерозпізнати вже використані дані *U* (накопичені).

3. Створити навчальний набір даних (фрагмент).

4. Навчити моделі p_i акустичного та мовного моделювання, використовуючи новий набір даних.

Цей процес продовжується до збіжності, досягнення певного обсягу набору даних, досягнення цілі WER або максимальної кількості ітерацій [5].

Результати показують, що конвеєр містить багато незалежних та важливих повторюваних кроків, які можуть бути часомісткими навіть для досвідчених інженерів. За допомогою ключових компонентів конвеєра, таких як DataProxy, Transformation, Data Input та Data Output, вдалося збільшити розмір APM набору даних з 267 до 2 500 годин за 1 289 годин за допомогою чотирьох GPU 1080ti та одного процесора AMD 3960x. Крім того, автоматичне кешування та інвалідація прискорюють експерименти та ітерації, дозволяючи переробку вихідних даних з додатковими кроками. Наприклад, хешування 10 000 годин аудіофайлів зайняло 25 хвилин зі сховищем типу SSD, тоді як використання типу HDD збільшило б час перевірки хешу. Альтернативним методом може бути використання часу зміни файлу, хоча це може бути менш надійним у деяких випадках, наприклад, при заміні відсутніх файлів на старіші версії.

З використанням поточного конвеєра нам вдалося досягти найкращого на ринку WER в розмірі 5,24 для української мови, і ми очікуємо, що подібний WER можна досягти для майже будь-якої мови з мінімум 250-годинним набором даних як стартовою точкою для використання поточного конвеєра [33].

РЕЗУЛЬТАТИ ПРОТОТИПУВАННЯ

В якості прототипів були обрані дві практичні задачі:

1. Кінцевий прототип програмного забезпечення для аналітики на основі голосових розмов за допомогою обробки природної мови.

2. Використання прототипу для оцінки емоційних станів учасників розмови.

Вимоги до прототипу

Прототип має бути ефективним та простим у використанні, з можливістю обробки як не в реальному часі, так і в реальному часі, і з розумінням архітектури для підвищення готовності до використання в продукції. Щоб досягти готових до використання результатів, ми повинні забезпечити стабільність і надавати повні результати, які включають не тільки розпізнавання мовлення, а й відокремлення осіб у моно- аудіо з особами в одному каналі, а також пунктуацію та інверсну нормалізацію тексту для перетворення слів на легко зрозумілі числа і символи (наприклад, «@», «#» тощо).

Майже всі завдання після опрацювання мають бути виконані після отримання результатів розпізнавання мовлення. Тому ми розробили конвеєр, який складається з наступних етапів:

1. Розпізнавання мовлення.
2. Діаризація (конвертування моно- в стерео-).
3. Нормалізація (конвертування слів в числа).
4. Пунктуація.

Алгоритм роботи прототипу

Нижче представлений алгоритм роботи прототипу:

1. Вхідний аудіофайл/стрім API POST / Websocket.
2. Вибір режим реального часу.
   2.1. В режимі реального часу:
      2.1.1. Розпізнати аудіо потік.
      2.1.2. Повернути результати в режимі реального часу через Websocket.
   2.2. В асинхронному режимі:
      2.2.1. Повернути ідентифікатор майбутніх результатів як відповідь API.
      2.2.2. Створити чергову задачу з розпізнаванням та післяобробкою конвеєру.
      2.2.3. Якщо розпізнавач готовий до нових задач:
         2.2.3.1. Розпізнати аудіофайл.
         2.2.3.2. Запустити обробку задач потпроцессінгу.
         2.2.3.3. Зберегти результати до бази даних та індексу.
3. Кінець.

Ми розробили архітектуру, показану на рис. 1, для виконання заданого алгоритму.

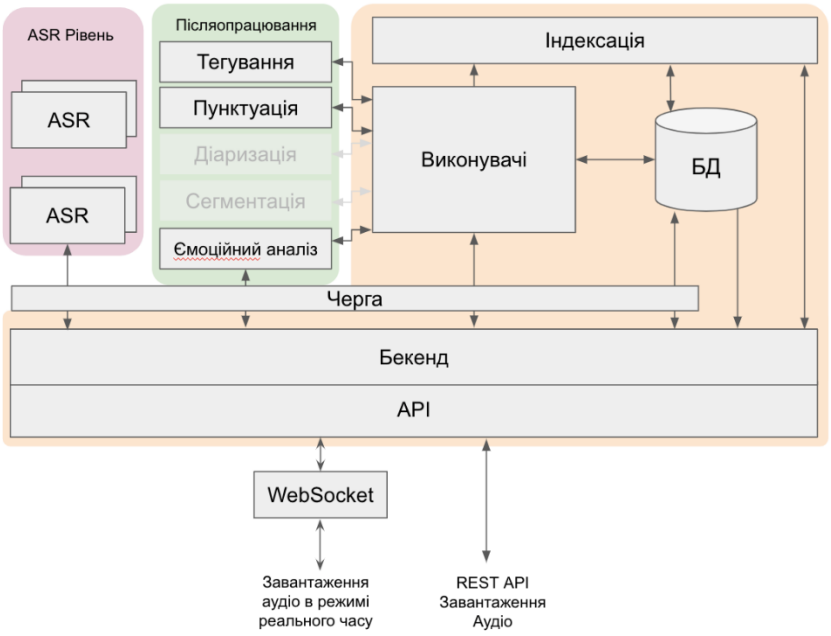

Рис. 1. Структурна схема прототипу

Швидкодія прототипу

Використовуючи прототип на сервері з 4xA100 GPU картами, ми обробили 45 000 годин аудіо за 24 години або 1 875 годин за 1 годину, що відповідає сучасним моделям [34–37]. Основними факторами, що впливають на високу ефективність, є фреймворки APM та WEB [38, 39].

Продуктивність моделі APM на CPU обмежена, великі моделі мають WER 4-6% і досягають 1 Real-Time Factor на 1 vCPU. Основним обмеженням продуктивності є передбачення APM, тому CPU-орієнтовані розгортання підходять для обмеженої кількості одночасних сесій (приблизно до 100 на найбільшому екземплярі AWS).

Обмеження прототипу

Під час тестування прототипу виникли обмеження та складності:

1. *Пріоритезація реального часу*: поточна версія прототипу не пріоритезує завдання в реальному часі над черговими, що означає, що немає впевненості, що завдання в реальному часі будуть прийняті, якщо черга та розпізнавачі мають багато роботи.

2. *Стабільність*: для підтримки як потокового, так і пакетного розпізнавання файлів (за допомогою одного APM сервера прототипу) поточна версія прототипу розпізнається шляхом потокового передачі аудіофайлів від працівника безпосередньо до APM, що не рекомендується для готового до використання вирішення та потребує доробки.

3. *Масштабованість*: поточна версія прототипу не має можливості балансування навантаження (ні для блоку API, ні для блоку APM та пост-обробки). Це повинно бути дороблено для готовності прототипу до використання вирішення.

4. *Зворотний зв'язок* та *видимість*: крім ID майбутніх результатів, користувачеві не надсилається жодна інформація про виконання завдання. Це може бути проблематичним, якщо сервіси зайняті, оскільки користувачі не зможуть точно знати, коли завдання буде завершено, і не отримають жодної інформації про оцінки та прогрес. Тому прототип повинен бути дороблений з додатковою видимістю та веб-хуками для отримання зворотного зв'язку, коли результати будуть готові та в разі будь-яких проблем з готовністю до використання вирішення [40].

Налаштування прототипу для аналізу емоцій

В якості прикладної задачі для перевірки прототипу було обрано задачу з розпізнавання мови. Прогрес у сфері автоматизованого розпізнавання мови значно прискорив автоматизацію контакт-центрів [41, 42]. Така автоматизація та покращення роботи людських операторів вимагає перекладу мовлення на текст для зрозуміння того, що було сказано та його настроїв [43]. Аналіз настроїв на основі тексту часто не може розпізнати гнів та радість, якщо люди не виражають їх через певні слова. Тим часом, люди кодують свої емоції в інтонаціях [44, 45]. Це створює потребу в стійкому розпізнаванні емоцій в мовленні як невід'ємній частині оцінки підтримки мережі.

Класифікація емоцій та оцінка емоційного залучення є одними з найбільш популярних потреб у кожному бізнесі. Те, як клієнти реагують, що викликає щастя і що спричиняє сум, створює безліч питань для багатьох компаній. Ми маємо на меті дослідити можливість стійкого багатомовного застосування визначення емоцій шляхом оцінки крос-мовної визначення емоцій.

Найпоширенішим способом та набором даних для розпізнавання емоцій є мультимодальні дані. Особливо ефективним для розпізнавання емоцій є відеодані, що є рідкісним випадком для виробничого середовища, такого як контакт-центр. У такому середовищі доступна тільки аудіоінформація для передбачення емоцій. Емоційні процеси, пов'язані з акустичними параметрами (частота, спектральна енергія, швидкість мовлення, shimmer та jitter). Крім того, хороші результати SER показують коефіцієнти мел-частотного кепстрального аналізу, спектральний звал, функціональний оператор Teager, спектрограми та ознаки форми голосу [46]. Поки що багато праці здійснюється у пошуку способу представлення акустичних сигналів та вилучення рухів у найкращий спосіб, але найбільш захоплююче запитання полягає у тому, де знайти відповідні дані для навчання моделей SER. Хоча для більшості розвинених мов у області NLP/мовлення немає проблеми з пошуком позначених даних у понад 200 годин, для більшості мов це все ще довгий шлях. У цій ситуації найочевиднішим підходом є навчання моделі за допомогою мови з багатим набором даних та застосування її до виробничого середовища з місцевою мовою. Це звучить добре, але на практиці працює далеко не добре. Ми маємо на меті оцінити переносимість моделей на різні мови як підхід та знайти найменші помилки, на які потрібно звернути увагу при цьому.

Вибір наборів даних з емоціями

В табл. 2 і 3 відображені наявні відкриті набори даних для тренування моделей розпізнавання емоцій [47–49]. Для наборі даних ми вибрали мови: китайську (ZN), німецьку (DE), естонську (ET), англійську (EN), фарсі (FA), французьку (FR) та урду (UR).

Таблиця 2

Порівняння наборів даних для аналізу основних емоцій

Мова	Набір даних	Нейтральний	Гнів	Смуток	Щастя	Здивування	Страх	Огида
ZN	ESD	+	+	+	+	+		
DE	EMODB	+	+	+	+		+	+
ET	EKORPUS	+	+	+	+			
EN	CREMA	+	+	+	+		+	+
EN	IEMOCAP	+	+	+	+	+	+	+
EN	RAVDESS	+	+	+	+	+	+	+
EN	SAVEE	+	+	+	+	+	+	+
EN	TESS	+	+	+	+	+	+	+
FA	ShEMO	+	+	+	+	+	+	
FR	OREAU	+	+	+	+	+	+	+
UR	URDU	+	+	+	+			

Таблиця 3

Перелік відкритих наборів даних для аналізу додаткових емоцій

Мова	Набір даних	Нудьга	Збудження	Розчарування	Спокій	Радість
DE	EMODB	+				
EN	IEMOCAP		+	+		
EN	RAVDESS				+	
EN	TESS					+

Для кожної мови ми створили окреме сховище (каталог) в наборі експериментів. Ми скоротили великі набори даних до порівнянного розміру, щоб збалансувати дані мов у рівній кількості годин. Для тестування та валідації ми вирішили використовувати 15% кожного мовного датасету, випадково вибрані між емоціями, що означає, що ми не брали 15% кожної емоції, а дали випадковий вибір прикладів для валідації. Кожну емоцію ми розмістили в сховищі (каталозі) з відповідним почуттям під відповідною мовою. Ми навчали 21 модель, по сім моделей для кожного набору емоцій [нейтральна-злість], [нейтральна-злість-сум], та [нейтральна-злість-сум-щастя].

Оцінка точності прототипу

Ми провели 147 експериментів, з яких 49 експериментів було проведено для кожного набору емоцій [нейтральний-злість], [нейтральний-злість-сум], і [нейтральний-злість-сум-щастя]. Кожен експеримент був повторений тричі, щоб оцінити відхилення точності. Стандартне відхилення оцінюється на 2%.

Після проведення числа експериментів ми можемо стверджувати, що передбачити три емоції значно важче, ніж дві, з медіанною точністю на 18% нижче по всіх мовах, тож варто звужувати тренування до мінімально необхідних класів. Ми також можемо побачити, наскільки важко передбачити чотири емоції порівняно з двома з медіанною точністю на 33% нижче по всіх мовах. Китайська мова не передається на будь-яку мову, навіть для такого простого налаштування, як дві емоції. Ми також можемо побачити неочікувану стійкість передачі моделі, навченої на фарсі. Ми також відстежуємо неочікувану поведінку віддзеркалення. Аналіз на наступних двох парах для двох емоцій DE-FA та FR-ZN, приводить нас до цікавих висновків, що якщо емоції добре передаються з джерелової мови на цільову мову, це не означає, що емоції можуть бути так само добре передані в зворотному напрямку між мовами [50].

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

В роботі представлений огляд методи розпізнавання мови а також методи обробки природньої мови. Окремо розглянуто останні технологічні прориви і досягнення областей.

Окремим блоком запроновані методи підготовки даних для використанні в тренуванні систем розпізнавання мови і конвейер автоматичного тренування систем розпізнавання мови використовуючи псевдо розмітку аудіо даних.

Прототип і вирішення реальної бізнес задачі з виявлення емоцій демонструють можливості і обмеження систем розпізнавання сови та емоційних станів. З використанням запропонованих методів псевдо-лейбування вдається без значних інвестицій в обчислювальні ресурси отримати точність розпізнавання близьку до лідерів ринку а для мов з незначною кількістю відкритих даних навіть перевершити.

План на подальше дослідження є поліпшення конвейеру автоматичної підготовки даних а також тренування моделей в розрізі швидкості і точності.

ПОДЯКА

Автор даної публікації висловлює подяку Ender Turing OÜ за визначення бізнес-проблеми, коментарі, виправлення, натхнення та обчислювальні ресурси.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Vaswani, A., *et al.* (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. http://arxiv.org/abs/1706.03762
2 Firth, J. R. (1957). A Synopsis of Linguistic Theory, 1930–1955.
3 Luscher, C., *et al.* (2019) RWTH ASR Systems for LibriSpeech: Hybrid vs Attention—w/o Data Augmentation, 1–5. http://arxiv.org/abs/1905.03072
4 Devlin, J., *et al.* (2019). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805
5 Xu, Q., *et al.* (2020). Iterative Pseudo-Labeling for Speech Recognition, 1–13. https://arxiv.org/abs/2005.09267
6 Mikolov, T., *et al.* (2013). Efficient Estimation of Word Representations in Vector Space. In *First International Conference on Learning Representations* (pp. 1–13). http://arxiv.org/abs/1301.3781
7 Peters, M., *et al*. (2018). Deep Contextualized Word Representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1* (pp. 2227–2237). https://doi.org/10.18653/v1/n18-1202
8 Cho, K., *et al.* (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. https://doi.org/10.3115/v1/w14-4012
9 Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. https://doi.org/10.1038/323533a0
10 Goodfellow, I., Bengio, Y., Courville, A. (2016). Sequence modeling: recurrent and recursive nets, *Deep Learning*, 367–415.
11 Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9(8)*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
12 Tanaka, T., *et al.* (2019). A Joint End-to-End and DNN-HMM Hybrid Automatic Speech Recognition System with Transferring Sharable Knowledge. *Interspeech*, 2210–2214. https://doi.org/10.21437/interspeech.2019-2263
13 Wang, D., Wang, X., Lv, S. (2019). An Overview of End-to-End Automatic Speech Recognition. *Symmetry, 11(8)*, 1018. https://doi.org/10.3390/sym11081018
14 Iosifov, I., Iosifova, O., Sokolov, V. (2020). Sentence Segmentation from Unformatted Text using Language Modeling and Sequence Labeling Approaches. In *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology*. https://doi.org/10.1109/picst51311.2020.9468084
15 Graves, A., *et al.* (2006). Connectionist Temporal Classification. In *23rd International Conference on Machine Learning* (pp. 369–376). https://doi.org/10.1145/1143844.1143891
16 Hsiao, R. (2020). Online Automatic Speech Recognition with Listen, Attend and Spell Model, 1–5. http://arxiv.org/abs/2008.05514
17 McDermott, E. (2018). A deep generative acoustic model for compositional automatic speech recognition, In *32nd Conference on Neural Information Processing Systems* (pp. 1–17).
18 Chan, W., et al. (2016). Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*. https://doi.org/10.1109/icassp.2016.7472621
19 Iosifova, O., *et al.* (2020). Techniques comparison for natural language processing. In *Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science, I(2631)* (pp. 57–67).
20 Iosifova, O., *et al.* (2021). Analysis of Automatic Speech Recognition Methods. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems* (pp. 252–257).
21 Radford, A., *et al.* (2019). Language Models Are Unsupervised Multitask Learners. *OpenAI Blog 1(8)*, 9.
22 Liu, Y., *et al.* (2019). Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/abs/1907.11692
23 Sanh, V., *et al.* (2019). DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *NeurIPS*. https://arxiv.org/abs/1910.01108
24 Yang, Z., *et al.* (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. https://arxiv.org/abs/1906.08237
25 Li, T., *et al.* (2021). A Short Study on Compressing Decoder-Based Language Modelshttps://arxiv.org/abs/2110.08460
26 Brown, T. B., *et al.* (2020). Language Models are Few-Shot Learners. https://arxiv.org/abs/2005.14165

27 Dai, Z., (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. https://arxiv.org/abs/1901.02860
28 Iosifov, I., *et al.* (2022). Natural Language Technology to Ensure the Safety of Speech Information. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems II* (pp. 216–226).
29 TensorFlow: The Functional API. https://www.tensorflow.org/guide/keras/functional
30 Apache Spark: ML Pipelines. https://spark.apache.org/docs/latest/ml-pipeline.html
31 Apache Airflow: DAGs. https://airflow.apache.org/docs/stable/concepts.html
32 Liao, H., McDermott, E., Senior, A. (2013). Large Scale Deep Neural Network Acoustic Modeling with Semi-Supervised Training Data for YouTube Video Transcription. In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (pp. 368–373). https://doi.org/10.1109/asru.2013.6707758
33 Romanovskyi, O., *et al.* (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. In *Advances in Computer Science for Engineering and Education IV* (pp. 25–36). https://doi.org/10.1007/978-3-030-80472-5_3
34 Georgescu, A.-L., *et al.* (2021). Performance vs. Hardware Requirements In State-of-the-Art Automatic Speech Recognition. *EURASIP Journal on Audio, Speech, and Music Processing, 2021(1)*. https://doi.org/10.1186/s13636-021-00217-4
35 Dutta, A., Ashishkumar, G., Rao, Ch. V. R. (2021). Improving the Performance of ASR System by Building Acoustic Models using Spectro-Temporal and Phase-Based Features. *Circuits, Systems, and Signal Processing, 41(3)*, 1609–1632. https://doi.org/10.1007/s00034-021-01848-w
36 Gondi, S., Pratap, V. (2021). Performance and Efficiency Evaluation of ASR Inference on the Edge. *Sustainability, 13(22)*, 12392. https://doi.org/10.3390/su132212392
37 Li, S., *et al.* (2019). Improving Transformer-Based Speech Recognition Systems with Compressed Structure and Speech Attributes Augmentation. *Interspeech, 2019*. https://doi.org/10.21437/interspeech.2019-2112
38 Kuchaiev, O., *et al.* (2019). NeMo: A Toolkit for Building AI Applications using Neural Modules, 36–44. https://arxiv.org/abs/1909.09577
39 Web Framework Benchmarks. https://www.techempower.com/benchmarks/#section=test&runid=7464e520-0dc2-473d-bd34-dbdfd7e85911
40 Pa Pa Win, H., Thu Thu Khine, P. (2020). Emotion Recognition System of Noisy Speech in Real World Environment. *International Journal of Image, Graphics and Signal Processing, 12(2)*, 1–8. https://doi.org/10.5815/ijigsp.2020.02.01
41 Kumar, J. A., Balakrishnan, M., Wan Yahaya, W. A. J. (2016). Emotional Design in Multimedia Learning: How Emotional Intelligence Moderates Learning Outcomes. *International Journal of Modern Education and Computer Science, 8(5)*, 54–63. https://doi.org/10.5815/ijmecs.2016.05.07
42 Dhar, P., Guha, S. (2021). A System to Predict Emotion from Bengali Speech. *International Journal of Mathematical Sciences and Computing, 7(1)*, 26–35. https://doi.org/10.5815/ijmsc.2021.01.04
43 Shirani, A., Nilchi, A. R. N. (2016). Speech Emotion Recognition based on SVM as Both Feature Selector and Classifier. *International Journal of Image, Graphics and Signal Processing, 8(4)*, 39–45. https://doi.org/10.5815/ijigsp.2016.04.05
44 Devi, J. S., Yarramalle, S., Prasad Nandyala, S. (2014). Speaker Emotion Recognition based on Speech Features and Classification Techniques. *International Journal of Image, Graphics and Signal Processing, 6(7)*, 61–77. https://doi.org/10.5815/ijigsp.2014.07.08
45 Lech, M., *et al.* (2020). Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding. *Frontiers in Computer Science, 2*. https://doi.org/10.3389/fcomp.2020.00014
46 Livingstone, S. R., Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE, 13(5)*, e0196391. https://doi.org/10.1371/journal.pone.0196391
47 Pichora-Fuller, M. K., Dupuis, K. (2020). Toronto emotional speech set (TESS) [Data set]. Borealis. https://doi.org/10.5683/SP2/E8H2MF
48 Desplanques, B., Thienpondt, J., Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *Interspeech, 2020*. https://doi.org/10.21437/interspeech.2020-2650
49 Iosifov, I., *et al.* (2022). Transferability Evaluation of Speech Emotion Recognition Between Different Languages. In *Advances in Computer Science for Engineering and Education* (pp. 413–426). https://doi.org/10.1007/978-3-031-04812-8_35
50 Romanovskyi, O., *et al.* (2022). Prototyping Methodology of End-to-End Speech Analytics Software. In *4th International Workshop on Modern Machine Learning Technologies and Data Science* (pp. 76–86).

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (19), 2023
ISSN 2663 - 4023

Ievgen A. Iosifov
Ph.D. student of the Department of Information and Cybersecurity named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv University, Kyiv, Ukraine
ORCID ID: 0000-0001-6203-9945
y.iosifov.asp@kubg.edu.ua

COMPLEX METHOD FOR AUTOMATIC RECOGNITION OF NATURAL LANGUAGE AND EMOTIONAL STATE

Abstract. Current trends in NLP emphasize universal models and learning from pre-trained models. This article explores these trends and advanced models of pre-service learning. Inputs are converted into words or contextual embeddings that serve as inputs to encoders and decoders. The corpus of the author's publications over the past six years is used as the object of the research. The main methods of research are the analysis of scientific literature, prototyping, and experimental use of systems in the direction of research. Speech recognition players are divided into players with huge computing resources for whom training on large unlabeled data is a common procedure and players who are focused on training small local speech recognition models on pre-labeled audio data due to a lack of resources. Approaches and frameworks for working with unlabeled data and limited computing resources are almost not present, and methods based on iterative training are not developed and require scientific efforts for development. The research aims to develop methods of iterative training on unlabeled audio data to obtain productively ready speech recognition models with greater accuracy and limited resources. A separate block proposes methods of data preparation for use in training speech recognition systems and a pipeline for automatic training of speech recognition systems using pseudo marking of audio data. The prototype and solution of a real business problem of emotion detection demonstrate the capabilities and limitations of owl recognition systems and emotional states. With the use of the proposed methods of pseudo-labeling, it is possible to obtain recognition accuracy close to the market leaders without significant investment in computing resources, and for languages with a small amount of open data, it can even be surpassed.

Keywords: automatic speech recognition; ASR; NLP; recurrent neural network; RNN.

REFERENCES

1 Vaswani, A., *et al.* (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. http://arxiv.org/abs/1706.03762
2 Firth, J. R. (1957). A Synopsis of Linguistic Theory, 1930–1955.
3 Luscher, C., *et al.* (2019) RWTH ASR Systems for LibriSpeech: Hybrid vs Attention—w/o Data Augmentation, 1–5. http://arxiv.org/abs/1905.03072
4 Devlin, J., *et al.* (2019). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805
5 Xu, Q., *et al.* (2020). Iterative Pseudo-Labeling for Speech Recognition, 1–13. https://arxiv.org/abs/2005.09267
6 Mikolov, T., *et al.* (2013). Efficient Estimation of Word Representations in Vector Space. In *First International Conference on Learning Representations* (pp. 1–13). http://arxiv.org/abs/1301.3781
7 Peters, M., *et al*. (2018). Deep Contextualized Word Representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1* (pp. 2227–2237). https://doi.org/10.18653/v1/n18-1202
8 Cho, K., *et al.* (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. https://doi.org/10.3115/v1/w14-4012
9 Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. https://doi.org/10.1038/323533a0
10 Goodfellow, I., Bengio, Y., Courville, A. (2016). Sequence modeling: recurrent and recursive nets, *Deep Learning*, 367–415.

11 Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9(8)*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
12 Tanaka, T., *et al.* (2019). A Joint End-to-End and DNN-HMM Hybrid Automatic Speech Recognition System with Transferring Sharable Knowledge. *Interspeech*, 2210–2214. https://doi.org/10.21437/interspeech.2019-2263
13 Wang, D., Wang, X., Lv, S. (2019). An Overview of End-to-End Automatic Speech Recognition. *Symmetry, 11(8)*, 1018. https://doi.org/10.3390/sym11081018
14 Iosifov, I., Iosifova, O., Sokolov, V. (2020). Sentence Segmentation from Unformatted Text using Language Modeling and Sequence Labeling Approaches. In *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology*. https://doi.org/10.1109/picst51311.2020.9468084
15 Graves, A., *et al.* (2006). Connectionist Temporal Classification. In *23rd International Conference on Machine Learning* (pp. 369–376). https://doi.org/10.1145/1143844.1143891
16 Hsiao, R. (2020). Online Automatic Speech Recognition with Listen, Attend and Spell Model, 1–5. http://arxiv.org/abs/2008.05514
17 McDermott, E. (2018). A deep generative acoustic model for compositional automatic speech recognition, In *32nd Conference on Neural Information Processing Systems* (pp. 1–17).
18 Chan, W., et al. (2016). Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*. https://doi.org/10.1109/icassp.2016.7472621
19 Iosifova, O., *et al.* (2020). Techniques comparison for natural language processing. In *Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science, I(2631)* (pp. 57–67).
20 Iosifova, O., *et al.* (2021). Analysis of Automatic Speech Recognition Methods. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems* (pp. 252–257).
21 Radford, A., *et al.* (2019). Language Models Are Unsupervised Multitask Learners. *OpenAI Blog 1(8)*, 9.
22 Liu, Y., *et al.* (2019). Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/abs/1907.11692
23 Sanh, V., *et al.* (2019). DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *NeurIPS*. https://arxiv.org/abs/1910.01108
24 Yang, Z., *et al.* (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. https://arxiv.org/abs/1906.08237
25 Li, T., *et al.* (2021). A Short Study on Compressing Decoder-Based Language Modelshttps://arxiv.org/abs/2110.08460
26 Brown, T. B., *et al.* (2020). Language Models are Few-Shot Learners. https://arxiv.org/abs/2005.14165
27 Dai, Z., (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. https://arxiv.org/abs/1901.02860
28 Iosifov, I., *et al.* (2022). Natural Language Technology to Ensure the Safety of Speech Information. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems II* (pp. 216–226).
29 TensorFlow: The Functional API. https://www.tensorflow.org/guide/keras/functional
30 Apache Spark: ML Pipelines. https://spark.apache.org/docs/latest/ml-pipeline.html
31 Apache Airflow: DAGs. https://airflow.apache.org/docs/stable/concepts.html
32 Liao, H., McDermott, E., Senior, A. (2013). Large Scale Deep Neural Network Acoustic Modeling with Semi-Supervised Training Data for YouTube Video Transcription. In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (pp. 368–373). https://doi.org/10.1109/asru.2013.6707758
33 Romanovskyi, O., *et al.* (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. In *Advances in Computer Science for Engineering and Education IV* (pp. 25–36). https://doi.org/10.1007/978-3-030-80472-5_3
34 Georgescu, A.-L., *et al.* (2021). Performance vs. Hardware Requirements In State-of-the-Art Automatic Speech Recognition. *EURASIP Journal on Audio, Speech, and Music Processing, 2021(1)*. https://doi.org/10.1186/s13636-021-00217-4
35 Dutta, A., Ashishkumar, G., Rao, Ch. V. R. (2021). Improving the Performance of ASR System by Building Acoustic Models using Spectro-Temporal and Phase-Based Features. *Circuits, Systems, and Signal Processing, 41(3)*, 1609–1632. https://doi.org/10.1007/s00034-021-01848-w
36 Gondi, S., Pratap, V. (2021). Performance and Efficiency Evaluation of ASR Inference on the Edge. *Sustainability, 13(22)*, 12392. https://doi.org/10.3390/su132212392
37 Li, S., *et al.* (2019). Improving Transformer-Based Speech Recognition Systems with Compressed Structure and Speech Attributes Augmentation. *Interspeech, 2019*. https://doi.org/10.21437/interspeech.2019-2112

38 Kuchaiev, O., *et al.* (2019). NeMo: A Toolkit for Building AI Applications using Neural Modules, 36–44. https://arxiv.org/abs/1909.09577
39 Web Framework Benchmarks. https://www.techempower.com/benchmarks/#section=test&runid=7464e520-0dc2-473d-bd34-dbdfd7e85911
40 Pa Pa Win, H., Thu Thu Khine, P. (2020). Emotion Recognition System of Noisy Speech in Real World Environment. *International Journal of Image, Graphics and Signal Processing, 12(2)*, 1–8. https://doi.org/10.5815/ijigsp.2020.02.01
41 Kumar, J. A., Balakrishnan, M., Wan Yahaya, W. A. J. (2016). Emotional Design in Multimedia Learning: How Emotional Intelligence Moderates Learning Outcomes. *International Journal of Modern Education and Computer Science, 8(5)*, 54–63. https://doi.org/10.5815/ijmecs.2016.05.07
42 Dhar, P., Guha, S. (2021). A System to Predict Emotion from Bengali Speech. *International Journal of Mathematical Sciences and Computing, 7(1)*, 26–35. https://doi.org/10.5815/ijmsc.2021.01.04
43 Shirani, A., Nilchi, A. R. N. (2016). Speech Emotion Recognition based on SVM as Both Feature Selector and Classifier. *International Journal of Image, Graphics and Signal Processing, 8(4)*, 39–45. https://doi.org/10.5815/ijigsp.2016.04.05
44 Devi, J. S., Yarramalle, S., Prasad Nandyala, S. (2014). Speaker Emotion Recognition based on Speech Features and Classification Techniques. *International Journal of Image, Graphics and Signal Processing, 6(7)*, 61–77. https://doi.org/10.5815/ijigsp.2014.07.08
45 Lech, M., *et al.* (2020). Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding. *Frontiers in Computer Science, 2*. https://doi.org/10.3389/fcomp.2020.00014
46 Livingstone, S. R., Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE, 13(5)*, e0196391. https://doi.org/10.1371/journal.pone.0196391
47 Pichora-Fuller, M. K., Dupuis, K. (2020). Toronto emotional speech set (TESS) [Data set]. Borealis. https://doi.org/10.5683/SP2/E8H2MF
48 Desplanques, B., Thienpondt, J., Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *Interspeech, 2020*. https://doi.org/10.21437/interspeech.2020-2650
49 Iosifov, I., *et al.* (2022). Transferability Evaluation of Speech Emotion Recognition Between Different Languages. In *Advances in Computer Science for Engineering and Education* (pp. 413–426). https://doi.org/10.1007/978-3-031-04812-8_35
50 Romanovskyi, O., *et al.* (2022). Prototyping Methodology of End-to-End Speech Analytics Software. In *4th International Workshop on Modern Machine Learning Technologies and Data Science* (pp. 76–86).