КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ ІМЕНІ БОРИСА ГРІНЧЕНКА

Кваліфікаційна наукова
праця на правах рукопису

ІОСІФОВ ЄВГЕН АНАТОЛІЙОВИЧ

УДК 004.056:004.934

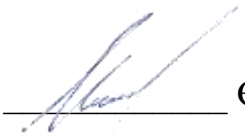
ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕЧНОГО РОЗПІЗНАВАННЯ ТА ПАРАМЕТРИЗАЦІЇ РЕЗУЛЬТАТІВ ОБРОБКИ ГОЛОСОВОЇ ІНФОРМАЦІЇ

Спеціальність 125 Кібербезпека
Галузь знань 12 Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



_____ Є. А. Іосіфов

Науковий керівник:
Соколов Володимир Юрійович
кандидат технічних наук, доцент

Київ – 2024

АНОТАЦІЯ

Іосіфов Є. А. Методи та засоби забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 125 Кібербезпека. – Київський столичний університет імені Бориса Грінченка, МОН України, Київ, 2024.

Дисертаційна робота присвячена вирішенню актуального наукового завдання, сутність якого полягає в підвищенні ефективності застосування безпечного розпізнавання та параметризації результатів обробки голосової інформації завдяки комбінуванню підходів розпізнавання природної мови та голосової інформації для побудови систем голосової автентифікації, виявлення намірів та визначення емоційного стану суб'єктів в інформаційно-комунікаційних системах (ІКС), а також впровадженню заходів управління кібербезпекою на державних підприємствах та в приватних організаціях.

Методологія обробки голосової інформації є потужним інструментом, який має значний вплив на безпеку держави та роботу комерційних організацій через автоматизацію процесів моніторингу електронних комунікацій та аудіоархівів, на основі розпізнавання в реальному часі мови, емоцій та намірів, чому сприяють декілька факторів, які змушують звернути увагу на методології, на актуальність їх удосконалення, а саме:

1. Зміна ландшафту кіберзагроз. Із появою генеративних моделей та збільшенням обчислювальних можливостей традиційні моделі безпеки, які покладаються на високо структуровані дані перестають адекватно виявляти та реагувати на підроблені аудіодані. Тому актуальними стають задачі по виявленню, реєстрації та реагуванню на нові виклики, а також швидкий розвиток даної галузі.

2. Перехід голосової інформації із телефонних розмов в телеконференції. При використанні традиційних телефонних переговорів до їхнього вмісту потенційно

мав доступ оператор зв’язку та державні органи. Тому тривалість та зміст розмов були меншими та піддавалися самоцензуруванню. Із переходом до телеконференцій вартість розмов зменшилася, а розповсюдження методів наскрізного шифрування створило уяву безпечності середовища, то абоненти стали вести більш відверті та довгі розмови, що стало особливо актуальним в епоху віддаленої роботи. Також через збільшення об’єму голосової інформації необхідно швидше зі сторони держави опрацьовувати її для вчасного виявлення, до прикладу, терористичних загроз, а зі сторони приватних підприємств – для виявлення витоку конфіденційних даних.

3. Порушення даних і зовнішні загрози. Діпфейки та введення спотворень в оригінальні аудіодані абонента створюють загрози для перенасичення інформаційної системи запитами. Виявлення та протидія фроду при аналізі намірів, в тому числі, генерації великої кількості фейкових намірів, призводять до перенавантаження зовнішніх пов’язаних системи та обмеженню ресурсів реагування, що створює загрозу недоотримання уваги легітимними суб’єктами.

4. Розширення ролі хмарних служб. Оскільки підприємства та організації все частіше використовують хмарні послуги для зберігання конфіденційних аудіоданих, то виникає потреба в додатковій обробці, в тому числі, деперсоналізації та видалення чутливих даних із аудіопотоку.

5. Вимоги відповідності. До персональних даних абонентів висуваються вимоги щодо їхньої конфіденційності в межах держаних стандартів (GDPR, HIPAA), комерційних (PCI DSS) та/або етичних обмежень. В свою чергу, аудіодані є важким видом інформації для структурованого пошуку та аналізу стосовно висунутих вимог та обмежень.

6. Безперервний моніторинг і адаптивна безпека. Обробка голосових даних може проводитися як архівних, так і в режимі реального часу, але вузьким місцем ІКС є потокова обробка даних. Тому реагування на інциденти може проводитися у два способи: невідкладні дії та розслідування інцидентів, але обидва підходи мають свій набір невирішених завдань.

7. Реагування на інциденти та виявлення загроз. Системи розпізнавання голосової інформації не мають в своєму складі механізмів щодо реагування на інциденти, тому повинні сигналізувати іншим системам в режимі реального часу. Інтеграція із зовнішніми ІКС для забезпеченні безпеки має обмеження на швидкодію та затримки на час обробки запитів, але все одно зменшує потенційну шкоду. Також слід зазначити, що актуальність реагування різко зменшується з плином часу.

Таким чином, дослідження щодо вдосконалення забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації є актуальним через його узгодження з поточним ландшафтом кібербезпеки, вирішення проблем, пов’язаних із віддаленою роботою, забезпеченням конфіденційності персональних даних абонентів та еволюцією природи кіберзагроз. Воно забезпечує адаптивний підхід до безпеки, необхідний для виявлення намірів та загроз аудіоінформації, яка циркулює в ІКС, телеконференціях та соціальних мережах.

Для досягнення мети в підвищенні ефективності застосування безпечного розпізнавання та параметризації результатів обробки голосової інформації було вирішено наступні задачі:

1. Вперше запропонований та математично обґрунтований метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів. При вирішенні завдання навчання на невеликій кількості нерозмічених даних реалізується підхід автоматичного отримання високоточного маркування, на відміну від існуючих методів навчання на великому об’ємі нерозмічених даних. Це дозволяє тренувати мовні моделі при наявності незначного обсягу аудіоданих, що значно знижує вартість формування тренувального набору даних порівняно з ручним і пришвидшує процес маркуванням щонайменше на 85%.

2. Вперше запропонований метод підвищення точності розпізнавання природної мови для близькоспоріднених мов. При вирішенні завдання розпізнавання природної мови фокус і увага концентруються саме на точності, на

відміну від існуючих методів розпізнавання, в яких основна увага приділяється якомога ширшому покриттю мов. Це дозволяє вбудовувати розроблений метод в системи ідентифікації про інциденти, в яких точність визначення природної мови впливає на їхній подальший аналіз, що, в свою чергу, підвищує точність роботи таких систем в середньому на 19,7% і мінімізує хибні спрацювання.

3. Вдосконалений метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей. Це дозволяє в подальшому використовувати розмічені на основі аудіоданих тексти та підвищити за рахунок цього ефективність підсистем розпізнавання мови та намірів.

4. Набув подальшого розвитку метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, що сукупно з методикою розпізнавання природної мови дає можливість більш точно визначати поріг емоційності для різних мов і тим самим мінімізувати нелегітимні спрацьовування в середньому на 18%. Також враховано рівень природної емоційності окремих народів, що дозволило відкалібрувати дані для впровадження заходів безпеки на державному рівні.

У вступі обґрунтовується важливість й актуальність теми дисертаційного дослідження, сформульовано мету та задачі роботи, визначено основні положення, наукову та практичну цінність отриманих результатів роботи та наведено особистий внесок автора.

У першому розділі здійснено аналіз існуючих методів розпізнавання та параметризації результатів обробки голосової інформації. Показаний еволюційний розвиток технологій роботи з природною мовою, проведено огляди технології розпізнавання природної мови та методів її обробки, також порівняльний аналіз мовних моделей та фреймворків. Визначено роль розпізнавання природної мови у забезпечення інформаційної безпеки підприємства, проаналізований поточний стан застосування методів автоматичного розпізнавання мови, визначено основні аспекти, підходи та принципи до навчання мовних моделей. Сформульовано актуально наукове завдання, яке полягає в подальшому розвитку методів вдосконалення безпечної роботи підприємства із аудіоінформацією на основі

розпізнавання та параметризації результатів обробки голосової інформації, зокрема програмних та організаційних аспектів їх забезпечення. Тому для його вирішення визначено мету роботи, яка полягає в підвищенні ефективності застосування безпечного розпізнавання та параметризації результатів обробки голосової інформації в ІКС завдяки комбінуванню підходів при формуванні розмічених аудіоданих для навчання мовних моделей, в процесі навчання та донавчання цих моделей.

У другому розділі визначено основні підходи до забезпечення безпеки голосової інформації. Вперше запропоновано метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів, що дало змогу запропонувати способи підвищення ефективності розпізнавання мовної інформації. Також зазначені обмеження та ризики при використанні методів розпізнавання голосової інформації в системах кібербезпеки. З урахуванням отриманих в поточному розділі результатів щодо розпізнавання багатомовних мовленнєвих емоцій та підходів щодо підвищення точності розпізнавання природної мови для близькоспоріднених мов в наступному розділі приділено увагу розширенню даних теоретичних обґрунтувань у вигляді практичної перевірки результатів.

У третьому розділі визначено ключові вимоги до даних для навчання мовних моделей. Запропонований вдосконалений метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей. Сформульовано проблеми та вибір підходів до її вирішення, верифіковано показники оцінювання та набори даних, а також проведене їхнє експериментальне дослідження. Вдосконалено метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами за допомогою підбору експериментальних аудіоданих, побудови алгоритму роботи та експериментальної установки, а також проведення експерименту та верифікації його результатів із розпізнання емоцій. Вперше запропоновано метод підвищення точності розпізнавання природної мови для близькоспоріднених мов. Для перевірки функціонування методу розроблено архітектура експериментальної установки, підібрано набори даних, визначено мови

із низькою точністю розпізнавання природної мови, проведено тренінг та верифікацію результатів експерименту із розпізнання двійок та трійок близькоспоріднених мов, а також проведено оцінку точності результатів для інших мовних пар за запропонованим методом.

Дисертація виконувалась в Київському столичному університеті імені Бориса Грінченка.

Результати наукових досліджень були використані на кафедрі інформаційної та кібернетичної безпеки імені професора Володимира Бурячка факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка в рамках науково-дослідної роботи: «Методи та моделі забезпечення кібербезпеки інформаційних систем переробки інформації та функціональної безпеки програмно-технічних комплексів управління критичної інфраструктури» (№ 0122U200483, КСУБГ, м. Київ).

Також результати наукових досліджень прийняті до впровадження в діяльність Київського столичного університету імені Бориса Грінченка (акт від 27.08.2024 року), «Ender Turing OÜ» (Таллінн, Естонія, акт від 07.09.2024 року) і «PP 2 SPV Limited Liability Company» (Ольштин, Польща, акт від 17.07.2024 року).

Ключові слова: кібербезпека, інформаційна безпека, голосова інформація, аудіодані, автентифікація, обробка природної мови, мовна модель, обробка тексту, автоматичне розпізнавання мовлення, штучний інтелект, машинне навчання, глибоке навчання, рекурентна нейронна мережа, глибока нейронна мережа, прихована марковська модель, енкодер, декодер, увага, трансформер, виявлення емоцій.

ANNOTATION

Iosifov I. A. Methods and Means of Ensuring Secure Recognition and Parameterization of Speech Information Processing Results. – Qualification of scientific work on the rights of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 125 Cybersecurity. – Borys Grinchenko Kyiv Metropolitan University, MES of Ukraine, Kyiv, 2024.

The dissertation is devoted to solving an urgent scientific problem, the essence of which is to increase the efficiency of applying secure recognition and parameterization of voice information processing results by combining natural language and voice information recognition approaches to build voice authentication systems, detect intentions and determine the emotional state of subjects in information and communication systems, as well as implement cybersecurity management measures at state-owned enterprises and in private.

The methodology of voice information processing is a powerful tool that has a significant impact on the security of the state and the work of commercial organizations through the automation of monitoring processes of electronic communications and audio archives, based on real-time recognition of speech, emotions, and intentions, which is facilitated by several factors that make us pay attention to the methodology and the relevance of their improvement:

1. The changing landscape of cyber threats. With the advent of generative models and increased computing power, traditional security models that rely on highly structured data no longer adequately detect and respond to fake audio data. Therefore, the tasks of detecting, registering, and responding to new challenges, as well as the rapid development of this industry, are becoming urgent.

2. Transition of voice information from telephone conversations to teleconferences. When traditional telephone conversations were used, the telecom operator and government agencies potentially had access to their content. Therefore, the duration and content of conversations were shorter and subject to self-censorship. With the transition

to teleconferencing, the cost of calls decreased, and the proliferation of end-to-end encryption methods created a perception of security, subscribers began to have more open and longer conversations, which became especially relevant in the era of remote work. Also, due to the increase in the volume of voice information, the state must process it faster to detect, for example, terrorist threats, and for private enterprises to detect leaks of confidential data.

3. Data breaches and external threats. Deepfakes and the introduction of distortions in the original audio data of a subscriber pose a threat of oversaturation of the information system with requests. Detecting and counteracting fraud in intent analysis, including the generation of a large number of fake intentions, leads to the overloading of externally connected systems and limiting response resources, which poses a threat of not receiving attention from legitimate actors.

4. Expanding the role of cloud services. As businesses and organizations increasingly use cloud services to store confidential audio data, there is a need for additional processing, including depersonalization and removal of sensitive data from the audio stream.

5. Compliance requirements. The personal data of subscribers is subject to confidentiality requirements within the framework of governmental standards (GDPR, HIPAA), commercial (PCI DSS), and/or ethical restrictions. Audio data, in turn, is a difficult type of information to search and analyze in a structured way due to the requirements and restrictions.

6. Continuous monitoring and adaptive security. Voice data can be processed both archived and in real-time, but the bottleneck of information and communication systems is streaming data processing. Therefore, incident response can be carried out in two ways: immediate actions and incident investigation, but both approaches have their own set of unresolved issues.

7. Incident response and threat detection. Voice recognition systems do not have incident response mechanisms, so they must signal other systems in real time. Integration with external information and communication systems for security has limitations on

performance and delays in processing requests, but still reduces potential damage. It should also be noted that the relevance of the response decreases dramatically over time.

Thus, the study on improving the secure recognition and parameterization of voice information processing results is relevant due to its alignment with the current cybersecurity landscape, addressing the problems associated with remote work, ensuring the confidentiality of subscribers' data, and the evolution of the nature of cyber threats. It provides the adaptive security approach needed to identify intent and threats to audio information circulating in information and communication systems, teleconferencing, and social media.

To achieve the goal of increasing the efficiency of secure recognition and parameterization of voice information processing results, the following tasks were solved:

1. For the first time, an automated pipeline method for creating training datasets from unlabeled audio recordings is proposed and mathematically justified. When solving the problem of training on a small amount of unlabeled data, an approach to automatically obtaining highly accurate labeling is implemented, unlike existing methods of training on a large amount of unlabeled data. This makes it possible to train speech models with a small amount of audio data, which significantly reduces the cost of generating a training dataset compared to manual training and speeds up the labeling process by at least 85%.

2. For the first time, a method for improving the accuracy of natural language recognition for closely related languages is proposed. When solving the task of natural language recognition, the focus and attention are concentrated on accuracy, unlike existing recognition methods that focus on the widest possible coverage of languages. This makes it possible to integrate the developed method into incident identification systems where the accuracy of natural language detection affects their further analysis, which, in turn, increases the accuracy of such systems by an average of 19.7% and minimizes false positives.

3. The improved method for segmenting unformatted text using language modeling and sequence labeling. This makes it possible to further use texts labeled based on audio data and thus increase the efficiency of speech and intent recognition subsystems.

4. The method of recognizing multilingual emotions was further developed by assessing the transfer between different languages, which, together with the natural language recognition method, makes it possible to more accurately determine the threshold of emotionality for different languages and thereby minimize illegitimate triggering by an average of 18%. The level of natural emotionality of individual people was also taken into account, which allowed us to calibrate the data for implementing security measures at the state level.

The introduction substantiates the importance and relevance of the topic of the dissertation research, formulates the purpose and objectives of the work, identifies the main provisions, and scientific and practical value of the results obtained, and presents the author's contribution.

The first section analyzes existing methods for recognizing and parameterizing the results of voice information processing. The evolutionary development of natural language technologies is shown, reviews of natural language recognition technology and methods of its processing are carried out, as well as a comparative analysis of language models and frameworks. The role of natural language recognition in ensuring the information security of an enterprise is defined, the current state of application of automatic speech recognition methods is analyzed, and the main aspects, approaches, and principles for training language models are identified. The article formulates an urgent scientific task, which consists of further development of methods for improving the secure operation of an enterprise with audio information based on recognition and parameterization of the results of voice information processing, in particular, the software and organizational aspects of their provision. Therefore, to solve this problem, the work aims to increase the efficiency of using secure recognition and parameterization of the results of voice information processing in information and communication systems by combining approaches to the formation of marked audio data for training language models, in the process of training and retraining these models.

The second section identifies the main approaches to ensuring the security of voice information. For the first time, an automated pipeline method was proposed for creating training datasets from unlabeled audio recordings, which made it possible to suggest ways

to improve the efficiency of speech information recognition. The limitations and risks of using voice information recognition methods in cybersecurity systems are also indicated. Taking into account the results obtained in the current section on the recognition of multilingual speech emotions and approaches to improving the accuracy of natural language recognition for closely related languages, the next section focuses on expanding these theoretical justifications in the form of practical verification of the results.

The third section defines the key data requirements for training language models. We propose an improved method for segmenting unformatted text using language modeling and sequence labeling. The problems and the choice of approaches to its solution are formulated, the evaluation indicators and data sets are verified, and their experimental study is carried out. The method for recognizing multilingual emotions is improved by assessing the transfer between different languages by selecting experimental audio data, building an algorithm and experimental setup, as well as conducting an experiment and verifying its emotion recognition results. For the first time, a method for improving the accuracy of natural language recognition for closely related languages is proposed. To verify the functioning of the method, the architecture of the experimental setup was developed, data sets were selected, languages with low natural language recognition accuracy were identified, training and verification of the results of the experiment on recognizing twos and threes of closely related languages were conducted, and the accuracy of the results for other language pairs using the proposed method was evaluated.

The dissertation was carried out at the Borys Grinchenko Kyiv Metropolitan University.

The results of scientific research were used at the Department of Information and Cybersecurity named after Professor Volodymyr Buriachok of the Faculty of Information Technologies and Mathematics of Borys Metropolitan Grinchenko Kyiv University within the framework of research work: “Methods and Models for Ensuring Cybersecurity of Information Systems, Information Processing and Functional Security of Software and Hardware Complexes for Critical Infrastructure Management” (No. 0122U200483, BGKMU, Kyiv).

Also, the results of scientific research have been accepted for implementation in the activities of Borys Grinchenko Kyiv Metropolitan University, Ender Turing OÜ (Tallinn, Estonia), and PP 2 SPV Limited Liability Company (Olsztyn, Poland).

Keywords: cybersecurity, information security, voice information, audio data, authentication, natural language processing, language model, text processing, automatic speech recognition, artificial intelligence, machine learning, deep learning, recurrent neural network, deep neural network, hidden Markov model, encoder, decoder, attention, transformer, emotion detection.

Наукові статті, опубліковані у наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України:

1. **Іосіфов, Є.** (2023). Комплексний метод по автоматичному розпізнаванню природньої мови та емоційного стану. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, *3*(19), 146–164. https://doi.org/10.28925/2663-4023.2023.19.146164.

2. Марценюк, М., Козачок, В., Богданов, О., **Іосіфов, Є.**, & Бржевська, З. (2023). Аналіз методів виявлення дезінформації в соціальних мережах за допомогою машинного навчання. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* *2*(22), 148–155. https://doi.org/10.28925/2663-4023.2023.22.148155.

3. **Іосіфов, Є.**, & Соколов, В. (2024). Методи аналізу природної мови та застосування нейронних мереж в кібербезпеці. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, *4*(24), 398–414. https://doi.org/10.28925/2663-4023.2024.24.398414.

4. **Іосіфов, Є.**, & Соколов, В. (2024). Порівняльний аналіз методів, технологій, сервісів та платформ для розпізнавання голосової інформації в системах забезпечення інформаційної безпеки. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, *1*(25), 468–486. https://doi.org/10.28925/2663-4023.2024.25.468486.

Наукові публікації, у яких додатково висвітлено результати дисертації:

1. Romanovskyi, O., **Iosifov, I.**, Iosifova, O., Sokolov, V., Kipchuk, F., & Sukaylo, I. (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. *Lecture Notes on Data Engineering and Communications Technologies*, *83*, 25–36. https://doi.org/10.1007/978-3-030-80472-5_3 (Scopus).

2. Iosifova, O., **Iosifov, I.**, Sokolov, V., Romanovskyi, O., & Sukaylo, I. (2021). Analysis of Automatic Speech Recognition Methods. *In Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS), 2923*, 252–257. (Scopus).

3. **Iosifov, I.**, Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Natural Language Technology to Ensure the Safety of Speech Information. *In Workshop on Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II), 3187*(1), 216–226. (Scopus).

4. **Iosifov, I.**, Iosifova, O., Romanovskyi, O., Sokolov, V., & Sukailo, I. (2022). Transferability Evaluation of Speech Emotion Recognition Between Different Languages. *Lecture Notes on Data Engineering and Communications Technologies, 134*, 413–426. https://doi.org/10.1007/978-3-031-04812-8_35 (Scopus).

5. Romanovskyi, O., **Iosifov, I.**, Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Prototyping Methodology of End-to-End Speech Analytics Software. *In 4th International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS), 3312*, 76–86. (Scopus).

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІКС –	інформаційно-комунікаційна система
ШНМ –	штучна нейронна мережа
ШІ –	штучний інтелект
ШПФ –	швидке перетворення Фур’є
API –	Application Programming Interface ‘прикладний програмний інтерфейс’
ASR –	Automatic Speech Recognition ‘автоматичне розпізнавання мови’
BERT –	Bidirectional Encoder Representations from Transformers ‘двоспрямовані кодувальні представлення з трансформерів’
BLSTM –	Bidirectional Long Short-Term Memory ‘двоспрямована довга короткочасна пам’ять’
BRNN –	Bidirectional Recurrent Neural Network ‘двонаправлена рекурентна нейронна мережа’
CNN –	Convolutional Neural Network ‘згорткова нейронна мережа’
CTC –	Connectionist Temporal Classification ‘коннекціоністська часова класифікація’
CV –	Common Voice ‘спільний голос’
DL –	Deep Learning ‘глибоке навчання’
DNN –	Deep Neural Network ‘глибока нейронна мережа’
EOS –	End of Sentence Token ‘маркер кінця речення’
GPT –	Generative Pre-trained Transformer ‘породжувальний попередньо тренований трансформер’
GPU –	Graphics Processing Unit ‘графічний процесор’
GRU –	Gated Recurrent Units ‘вентильний рекурентний вузол’
HMM –	Hidden Markov Model ‘прихована марковська модель’
HMM-GMM –	Hidden Markov Model with Gaussian Mixture Emissions ‘прихована марковська модель з гауссовою сумішшю викидів’

IDE – Integrated Development Environment ‘інтегроване середовище розробки’

LAS – Listen, Attend, and Spell ‘слухайте, відвідуйте та промовляйте’

LDC – Linguistic Data Consortium ‘консорціум лінгвістичних даних’

LSTM – Long Short-Term Memory ‘довга короткочасна пам’ять’

MFCC – Mel-Frequency Cepstral Coefficients ‘мелчастотні кепстральні коефіцієнти’

ML – Machine Learning ‘машинне навчання’

NLP – Natural Language Processing ‘обробка природної мови’

NLT – Natural Language Technology ‘технологія природної мови’

RNN – Recurrent Neural Network ‘рекурентна нейронна мережа’

SER – Speech Emotion Recognition ‘розпізнавання мовних емоцій’

SLI – Sign Language Interpreting ‘переклад мови жестів’

SLR – Sign Language Recognition ‘розпізнавання мови жестів’

SPT – Speech Processing Toolkit ‘інструментарій для обробки мовлення’

STT – Speech-to-Text ‘генерація тексту із мовлення’

TTS – Text-to-Speech ‘генерації мовлення з тексту’

WER – Word Error Rate ‘частота помилок у словах’

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний світ характеризується стрімким розвитком інформаційних технологій, які суттєво впливають на всі сфери суспільства, зокрема на організаційну та національну безпеку. Водночас, зростання обсягів цифрових комунікацій та інформаційних потоків створює нові виклики для організацій і державних органів у сфері збору та аналізу даних. Однак, дане завдання ускладнюється через величезні масиви інформації, які надходять у вигляді дзвінків, голосових повідомлень та текстових даних, що потребує ефективних методів обробки. Причиною такого розповсюдження є людський природний спосіб комунікації – мовлення. Така трансформація призвела до необхідності впровадження автоматизованих систем аналізу, що, в свою чергу, призводить до підвищення ефективності реагування на потенційні загрози.

Кількість і вектори кіберзагроз та інформаційних атак постійно зростає. Цьому зокрема сприяло підвищення інтересу до технологій розпізнавання голосової мови та обробки природної мови (від англ. Natural Language Processing, NLP) з боку зловмисників, і як відповідь з боку державних установ та організацій, як інструментів для забезпечення безпеки. Тому безпечне NLP означає процес обробки та інтерпретації мовлення або тексту за допомогою технологій штучного інтелекту з дотриманням вимог безпеки і приватності, що включає захист аудіоданих від витоків, несанкціонованого доступу, дотримання конфіденційності суб’єктів, спотворення змісту або маніпуляцій, а також із урахуванням правових та етичних аспектів.

В результаті, виникає необхідність у впровадженні сучасних технологій розпізнавання мови та NLP в системи організаційної і державної безпеки. Водночас, існують технічні, етичні та правові аспекти, які потребують детального аналізу. Однак, в зв’язку з важливістю балансу між забезпеченням безпеки та удосконалення процедури розпізнавання природньої мови, ці питання набувають особливої актуальності.

У даному контексті, розгляд та розуміння різних підходів, методів та сучасних практик застосування технологій розпізнавання мови стають важливими. Так,

одним із найперспективніших підходів вважається використання методів машинного навчання та глибоких нейронних мереж для підвищення точності розпізнавання та аналізу мовних даних.

Основний принцип цих технологій полягає в здатності систем навчатися на великих обсягах даних і можливості дотренування в майбутньому. Іншими словами, чим більше даних обробляється, тим точнішою можна зробити систему. Завдяки використанню таких технологій, можливо автоматизувати процеси моніторингу та аналізу, що є критично важливим для своєчасного виявлення та реагування на потенційні загрози. Все це можливо завдяки високоточним сучасним системам автоматичного розпізнавання мовлення (від англ. Automatic Speech Recognition, ASR). Водночас, слід відзначити, що існують ризики, пов’язані з можливими помилками в розпізнаванні або зловживаннями цими технологіями. Останні дослідження в галузі голосових інтерфейсів дають нам уявлення про те, що в майбутньому проблема величезного розмаїття користувацьких інтерфейсів буде вирішена. Саме тому, останнім часом стрімко набирає вагомості питання застосування технологій розпізнавання мови та NLP з урахуванням етичних та правових норм у контексті забезпечення організаційної та державної безпеки.

Дослідженням даного питання займається досить велика кількість вчених, серед яких: John Rupert Firth, John Joseph Hopfield, Ashish Vaswani, Yoshua Bengio, Geoffrey Hinton, Kyung Hyun Cho, David Rumelhart, Ronald Williams, Sepp Hochreiter, Jürgen Schmidhuber, Leonard Baum, Ted Petrie, Lawrence Rabiner, Biing Hwang Juang, Abdel-rahman Mohamed, Junyoung Chung, Caglar Gulcehre та інші. Переважна більшість робіт присвячена розробці систем машинного навчання і лінгвістичних моделей для переводу аудіоданих в текст та навпаки, а також по розпізнаванню емоційного стану суб’єктів, в тому числі, для підвищення рівня захищеності ІКС, в яких обробляються аудіодані. Крім того, розглядаються не лише теоретичні засади, але показана ефективність впровадження на державних та приватних підприємствах та організаціях. Водночас, більшість уваги зосереджується саме на практичних результатах, про що свідчить велика кількість

наукових робіт, які видаються лише у вигляді препринтів, але це не зважає цим роботам впливати на динамічний розвиток даної досить молодої галузі знань.

Таким чином, з приведеного аналізу можна зробити висновок, що в практиці застосування концептуальних принципів безпечного розпізнавання та параметризації результатів обробки голосової інформації на підприємствах критичної інфраструктури та в державних органах загострилося протиріччя між забезпеченням безпеки та удосконаленням процедури розпізнавання природньої мови (в голосовому її представленні) при навчанні нових та донавчанні існуючих мовних моделей, які б дозволили ефективніше визначати загрози в потоці аудіоданих.

У зв’язку з цим, існує необхідність вирішення актуального наукового завдання, сутність якого полягає в подальшому розвитку методів вдосконалення безпечного розпізнавання та параметризації результатів обробки голосової інформації, а також засобів для протидії виникаючим загрозам.

Зв’язок роботи з науковими програмами, планами, темами. Напрям дисертаційного дослідження безпосередньо пов’язаний з реалізацією доктрини інформаційної безпеки України, Стратегії інформаційної безпеки та Стратегії кібербезпеки України. Дисертаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Київського столичного університету імені Бориса Грінченка в рамках науково-дослідної роботи: «Методи та моделі забезпечення кібербезпеки інформаційних систем переробки інформації та функціональної безпеки програмно-технічних комплексів управління критичної інфраструктури» (№0122U200483, КСУБГ, м. Київ). Під час виконання дисертаційної роботи було отримано два індивідуальні гранти: «Research of Natural Language Processing. Stage 3» (№67-090122, 2022–2023) і «Research of Speech Emotion Recognition» (№89-080123, 2023–2024) на публікації матеріалів дослідження в наукових виданнях від організації «Ender Turing OÜ» (м. Таллінн, Естонія).

Мета і завдання дослідження. *Мета* дисертаційного дослідження полягає в підвищенні ефективності застосування безпечного розпізнавання та параметризації

результатів обробки голосової інформації в ІКС завдяки комбінуванню підходів при формуванні розмічених аудіоданих для навчання мовних моделей та в процесі навчання та донавчання цих моделей.

У відповідності до поставленої мети для вирішення наукового завдання в роботі визначено та розв’язано такі *часткові* ***завдання***:

– проаналізовано поточний стан і підходи до забезпечення безпеки голосової інформації, як одного із ключових елементів персональних даних суб’єкта, а також розглянуті сучасні архітектуру та структуру елементів ІКС, які працюють із аудіоданими;

– проведено детальний аналіз метрик природної мови та критеріїв для оцінювання якості її обробки;

– розроблено модель автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів та визначені критерії оцінки роботи цієї моделі;

– визначено способи підвищення ефективності розпізнавання мовної інформації при одночасній роботі із кількома мовами при визначенні емоційного стану суб’єкта;

– формалізовано переваги, обмеження, ризики та виклики при впровадженні та застосовано методів розпізнавання голосової інформації;

– сформульовано вимоги до даних для навчання мовних моделей та досліджено доступні мовні корпуси для української мови;

– покращено сегментацію неформатованого тексту з використанням мовного моделювання та маркування послідовностей;

– досліджено нові підходи до розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, а також запропоновано спосіб підготовки та валідації вхідних даних;

– запропоновано підходи до підвищення точності розпізнавання природної мови для близькоспоріднених мов;

– вибрано мови із низької точністю для проведення експериментів та проведено за допомогою них тренінг моделі, а також верифіковані результати експериментів.

Об'єктом дослідження є процес забезпечення безпеки голосової інформації та емоційного стану при побудові розподіленої ІКС.

Предметом дослідження є методи та засоби забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації як ключових елементів формування дієвої політики інформаційної безпеки підприємства або державної установи на основі реалізації базових положень щодо NLP.

Методи дослідження. Для проведення досліджень в дисертаційній роботі використовувалися методи порівняльного аналізу; теорія ймовірності та математичної статистики; критичний аналіз обмежень та ризиків застосування; технологія рекурентних нейронних мереж; архітектура енкодерів-декодерів і механіки для формування уваги; прихована і гібридна прихована марковські моделі; коннекціоністська модель часової класифікації; послідовна модель; методи трансформеру та конвеєру для безперервних потоків аудіоданих; методи валідації експериментальних результатів; методи моделювання систем управління інформаційною безпекою; етичні обмеження.

Наукова новизна одержаних результатів полягає в подальшому розвитку теоретичних і практичних методів та засобів забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації:

1. Вперше запропонований та математично обґрунтований метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів. При вирішенні завдання навчання на невеликій кількості нерозмічених даних реалізується підхід автоматичного отримання високоточного маркування, на відміну від існуючих методів навчання на великому об'ємі нерозмічених даних. Це дозволяє тренувати мовні моделі при наявності незначного обсягу аудіоданих, що значно знижує вартість формування

тренувального набору даних порівняно з ручним і пришвидшує процес маркуванням щонайменше на 85%.

2. Вперше запропонований метод підвищення точності розпізнавання природної мови для близькоспоріднених мов. При вирішенні завдання розпізнавання природної мови фокус і увага концентруються саме на точності, на відміну від існуючих методів розпізнавання, в яких основна увага приділяється якомога ширшому покриттю мов. Це дозволяє вбудовувати розроблений метод в системи ідентифікації про інциденти, в яких точність визначення природної мови впливає на їхній подальший аналіз, що, в свою чергу, підвищує точність роботи таких систем в середньому на 19,7% і мінімізує хибні спрацювання.

3. Вдосконалений метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей. Це дозволяє в подальшому використовувати розмічені на основі аудіоданих тексти та підвищити за рахунок цього ефективність підсистем розпізнавання мови та намірів.

4. Набув подальшого розвитку метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, що сукупно з методикою розпізнавання природної мови дає можливість більш точно визначати поріг емоційності для різних мов і тим самим мінімізувати нелегітимні спрацьовування в середньому на 18%. Також враховано рівень природної емоційності окремих народів, що дозволило відкалібрувати дані для впровадження заходів безпеки на державному рівні.

Практичне значення одержаних результатів полягає в наступному: динамічний розвиток технологій розпізнавання природної мови призводить до появи можливостей розпізнавати та реагувати на аудіодані в режимі реального часу, але, в той самий час, виникають нові загрози для підприємств критичної інфраструктури, державних органів, приватного сектору та окремих громадян, що дозволяє зрозуміти актуальність і важливість безпечного розпізнавання та параметризації результатів обробки голосової інформації.

Саме ці тренди обумовлюють практичну значущість запропонованого в дослідженні принципового переходу до автоматизованого конвеєру для створення

навчальних наборів даних з нерозмічених аудіозаписів, який як найменше потребує людського втручання для ручної розмітки, а підвищення точності розпізнавання природної мови для близькоспоріднених мов дозволяє розширити кількість корпусів та перевикористовувати одні і ті самі корпуси для різних мов. Запропоноване рішення щодо сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей дозволило підвищити швидкодію та зекономити обчислювальні ресурси, а додатковий підхід до розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами дозволив створити комплексне рішення для забезпечення безпечної системи збору та обробки аудіоданих. Перспективність запропонованих рішень для таких галузей як прихованої обробки аудіоданих, інтерактивних голосових меню для служб спасіння, медичних служб, банківської сфери, маркетплейсів, транскрибування телеконференцій тощо є очевидною.

Апробація результатів дисертації. Основні теоретичні та практичні результати були представлені та обговорені в ході ряду наукових конференцій:

1. International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS), 2022.
2. International Conference on Computer Science, Engineering and Education Applications (ICCSEEA), 2021 і 2022.
3. Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS), 2021 і 2022.

Публікації. Основні результати дисертації висвітлено у 9 наукових публікаціях, із них 1 – одноосібна, 8 – у співавторстві: 4 статті (з них 3 у співавторстві) у наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України; 5 статей (з них усі у співавторстві) у періодичних наукових виданнях, проіндексованих в наукометричній базі даних Scopus. Наукові результати дисертації повною мірою висвітлено у наукових публікаціях.

Особистий внесок здобувача. Дисертація є самостійною науковою працею, в якій висвітлено власні ідеї і розробки автора, що дозволили вирішити поставлені

завдання. Робота містить теоретичні та методичні положення і висновки, сформульовані здобувачкою особисто. Використані в дисертації ідеї, положення чи гіпотези інших авторів мають відповідні посилання і використані лише для підкріплення ідей здобувача.

У статті «Комплексний метод по автоматичному розпізнаванню природньої мови та емоційного стану» опублікованій одноосібно, внесок Іосіфова Є.А. полягає у розробці комплексного методу із розпізнавання природної мови та емоційного стану, що загалом складає 100% тексту статті.

У статті «Аналіз методів виявлення дезінформації в соціальних мережах за допомогою машинного навчання» опублікованій у співавторстві, внесок Іосіфова Є.А. полягає в огляді існуючих підходів до виявлення фейкових новин з точки зору машинного навчання для забезпечення кібербезпеки, що загалом складає 30% тексту статті.

У статті «Методи аналізу природної мови та застосування нейронних мереж в кібербезпеці» опублікованій одноосібно, внесок Іосіфова Є.А. полягає у проведені аналізу існуючих методів розпізнавання природної мови та застосування їх в забезпеченні інформаційної безпеки, що загалом складає 80% тексту статті.

У статті «Порівняльний аналіз методів, технологій, сервісів та платформ для розпізнавання голосової інформації в системах забезпечення інформаційної безпеки» опублікованій у співавторстві, внесок Іосіфова Є.А. полягає у дослідженні і порівняння технологій, підходів, алгоритмів та платформ для розпізнавання голосової інформації та формування параметрів для забезпечення інформаційної безпеки, що загалом складає 50% тексту статті.

У статті «Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition» опублікованій у співавторстві, внесок Іосіфова Є.А. полягає у побудові моделі автоматизованого конвеєра для маркування нерозмічених даних для їхнього використання при тренуванні моделей, що загалом складає 40% тексту статті.

У статті «Analysis of Automatic Speech Recognition Methods» опублікованих у співавторстві, внесок Іосіфова Є.А. полягає у дослідженні і порівнянні підходів і

алгоритмів для розпізнавання природної мови, що загалом складає 45% тексту статті.

У статті «Natural Language Technology to Ensure the Safety of Speech Information» опублікованих у співавторстві, внесок Іосіфова Є.А. полягає у дослідженні алгоритмів та формуванні вимог до наборів аудіоданих для тренування моделей для роботи з голосовою інформацією та забезпечення їхньої безпеки, що загалом складає 65% тексту статті.

У статті «Transferability Evaluation of Speech Emotion Recognition Between Different Languages» опублікованій у співавторстві, внесок Іосіфова Є.А. полягає у побудові моделі переносу знання між мовними моделями для підвищення точності розпізнавання емоцій в голосовій інформації, що загалом складає 60% тексту статті.

У статті «Prototyping Methodology of End-to-End Speech Analytics Software» опублікованих у співавторстві, внесок Іосіфова Є.А. полягає у апробації методології на реальних задачах і у розробці прототипу системи розпізнавання і аналізу голосової інформації, що загалом складає 45% тексту статті.

Структура та обсяг дисертаційної роботи. Дисертація складається зі вступу, трьох розділів, висновків, списку використаних джерел із 175 найменування на 26 сторінках і 8 додатків. Загальний обсяг роботи становить 214 сторінок серед яких 179 сторінок основного тексту, 34 рисунки і 20 таблиць.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ТА ПАРАМЕТРИЗАЦІЇ РЕЗУЛЬТАТІВ ОБРОБКИ ГОЛОСОВОЇ ІНФОРМАЦІЇ

1.1. Еволюційний розвиток технологій роботи з природною мовою

Машинне навчання (від англ. Machine Learning, ML) та глибоке навчання (від англ. Deep Learning, DL) зазнали значного розвитку протягом останніх десятиліть. Від простих алгоритмів, заснованих на статистиці, до складних нейронних мереж, здатних розпізнавати обличчя та грати в ігри краще, ніж люди, розвиток цих технологій трансформував численні галузі. Нижче буде розглянута історія ML та DL, еволюція точності, математичні основи та деякі проблеми, які були вирішені завдяки цим технологіям.

1.1.1. Історія та розвиток машинного навчання

ML бере свій початок із ранніх робіт у галузі штучного інтелекту (ШІ) та статистики. У 1950-х роках Алан Тюрінг запропонував концепцію «машин, що навчаються». Перші алгоритми ML, такі як перцептрон Френка Розенблатта (1958) [1], були простими нейронними мережами, здатними вирішувати базові завдання класифікації.

У 1960–70-х роках були розроблені інші ключові алгоритми, такі як метод найменших квадратів, дерев’яні моделі рішень та алгоритми групування (кластери). Проте, обчислювальні обмеження тих часів не дозволяли масштабувати ці алгоритми для великих задач. У 1980-х роках з’явилося нове зацікавлення нейронними мережами завдяки винаходу алгоритму зворотного поширення помилки ‘back propagation’ для тренування багатошарових нейронних мереж. Це дозволило тренувати більш складні моделі, але все ще залишалися проблеми з обчислювальною потужністю та великими даними.

1.1.2. Трансформація машинного навчання в глибоке навчання

DL, підмножина ML, стало популярним на початку 2000-х років завдяки зростанню обчислювальних потужностей та доступності великих обсягів даних. Глибокі нейронні мережі мають кілька прихованих шарів, що дозволяє їм вивчати складні функції та закономірності.

Одним з ключових моментів в історії DL став 2012 рік, коли команда під керівництвом Джеффрі Хінтона виграла конкурс ImageNet, використовуючи глибоку згорткову нейронну мережу (від англ. Convolutional Neural Network, CNN) [2]. Цей успіх демонстрував, що DL може значно перевершити традиційні методи ML у задачах розпізнавання образів. Вслід за цим все більше розвитку стали набувати NLP і підходи роботи з текстом, що переросло у появу архітектури трансформерів ‘transformers’ і пізніше ChatGPT [3] на основі Generative Pretrained Transformers [4]. Можна виділити основні етапи розвитку DL, особливо в контексті NLP (табл. 1.1).

Таблиця 1.1

Етапи розвитку машинного та глибокого навчання

Роки	Подія
Поява ML	
1950-ті	Концепція «машин, що навчаються» Алана Тюрінга
1958	Перцептрон Френка Розенблата
1960-70-ті	Розвиток алгоритмів і методів найменших квадратів, дерев’яних моделей рішень та алгоритмів групування
1980-ті	Винахід алгоритму зворотного поширення помилки
Перехід до DL	
2000-ті	Поява DL завдяки зростанню обчислювальних потужностей та великих обсягів даних
2012	Успіх команди Джеффрі Хінтона на конкурсі ImageNet з використанням CNN
2017	Поява архітектури трансформерів
2018	Поява двонаправленої моделі – BERT (від англ. Bidirectional Encoder Representations from Transformers)
2018–2020	Перехід до великих лінгвістичних моделей GPT (від англ. Generative Pre-trained Transformer)
2022	Поява ChatGPT – одного з найвідоміших прикладів використання DL в NLP

ChatGPT базується на архітектурі трансформера і розроблений компанією OpenAI. Його метою є забезпечення природного та контекстуального спілкування з користувачами через текст.

1.1.3. Основні концепти і механізми машинного та глибинного навчання

ML базується на статистичних методах та оптимізації. Основні компоненти включають лінійну регресію (модель для передбачення залежної змінної на основі незалежних змінних), логістичну регресію (модель для задач класифікації) та метод найменших квадратів (алгоритм для знаходження найкращої відповідності моделі).

DL додає складності за рахунок використання багатошарових нейронних мереж. Основні компоненти включають:

– нейрон – базову обчислювальну одиницю, що приймає вхідні сигнали, застосовує вагові коефіцієнти та нелінійну функцію активації;

– зворотне поширення помилки (back propagation) – алгоритм для коригування ваг нейронної мережі;

– функції активації – сигмоїдальна, зрізаний лінійний вузол (ReLU), гіперболічна тангенс – додають нелінійність до моделей, що дозволяє вивчати складні шаблони;

– CNN – спеціалізовані для обробки зображень.

Можна виділити дві основні групи моделей DL для вирішення задач NLP: модель вбудовування слів (представлення слів у вигляді векторів, які захоплюють семантичні значення, наприклад, Word2Vec, GloVe), і модель послідовності.

В свою чергу, моделі послідовності поділяються на:

– рекурентна нейронна мережа (від англ. Recurrent Neural Network, RNN) – моделі для обробки послідовностей даних, таких як текст. Вони можуть обробляти послідовні дані, але мають проблеми з довгостроковими залежностями.

– вдосконалена RNN (або довга короткочасна пам’ять ‘Long Short-Term Memory’, LSTM) – вдосконалена версія RNN, що може зберігати довгострокову

пам’ять і краще підходить для обробки тексту та мовлення, а також краще справляються з довгостроковими залежностями;

– трансформери – архітектура, яка замінила RNN і LSTM для багатьох NLP задач завдяки своїй здатності обробляти послідовності паралельно та враховувати далекі залежності.

Трансформери у свою чергу спираються на додаткові механізми, такі як:

– механізм уваги ‘attention mechanism’, який дозволяє моделі фокусуватися на різних частинах вхідної послідовності, що значно покращує розуміння контексту покращуючи обробку довгих залежностей;

– BERT – двонаправлена модель, яка розуміє контекст слова з обох боків (зліва і справа);

– GPT – моделі для генерації тексту, здатні створювати зв’язний та контекстуально правильний текст.

1.1.4. Еволюція точності та складності вирішення задач

Підвищення точності алгоритмів ML та DL було досягнуто завдяки вдосконаленню методів, збільшенню обчислювальних потужностей та обсягів даних. Зокрема, використання графічних процесорів (від англ. Graphics Processing Unit, GPU) для тренування нейронних мереж значно прискорило цей процес.

DL дозволило вирішити численні задачі, які раніше вважалися нерозв’язними або дуже складними:

– розпізнавання образів – значне покращення в точності розпізнавання обличчя, об’єктів та сцен;

– NLP – розвиток моделей, таких як трансформери, що дозволили значно покращити розуміння та генерацію тексту;

– автоматизація, прискорення задач, які виконують люди і перехід до появи роботів-акторів, які можуть виконувати частку або повну задачу описану природною мовою, за рахунок роботи з голосом і текстом;

– ігри – створення систем, які перевершують людей у складних іграх, таких як шахи та Go (наприклад, AlphaGo [5]).

1.1.5. Сучасні проблеми та виклики використання нейронних мереж

Одна з головних проблем DL – інтерпретованість моделей. Глибокі нейронні мережі працюють як «чорні ящики», і важко зрозуміти, як вони приймають рішення. Це особливо критично в таких галузях, як медицина або фінанси, де необхідно пояснювати рішення моделей.

Тренування глибоких нейронних мереж вимагає великих обчислювальних ресурсів. Це створює бар'єри для дослідників та компаній з обмеженими ресурсами. Моделі DL можуть перенавчатися на тренувальних даних і погано працювати на нових, невідомих даних. Це проблема генералізації, яка потребує вдосконалення методів регуляризації та інших підходів.

ML та DL пройшли довгий шлях від своїх витоків до сучасних передових технологій. Завдяки поєднанню теоретичних досягнень, зростання обчислювальних потужностей та великих даних, ці технології дозволили вирішувати складні задачі, що раніше були нерозв'язними. Проте, залишаються виклики, такі як інтерпретованість, потреба в обчислювальних ресурсах та проблема генералізації, які вимагають подальшого дослідження та вдосконалення.

1.2. Огляд технології розпізнавання природної мови

Значні досягнення в галузі DL останнього десятиліття відкривають нові можливості та вимоги для бізнесу, урядів та громадян. Такі досягнення в технології природної мови (від англ. Natural Language Technology, NLT) створюють для бізнесу можливість автоматизувати більшість рутинних завдань у спілкуванні з клієнтами та спрямувати ресурси на більш цікаві та творчі завдання.

Дослідження [6] присвячене порівнянню основних підходів у NLP та розпізнаванні мови. Автори досліджують вимоги до наборів даних для тренування текстових та мовних моделей, порівнюють основні інструменти і техніки [7], а також описують останні тренди в цій сфері. А в роботі [8] розглядаються структури систем автоматичного розпізнавання мови, включаючи гібридні та кінцеві моделі. Описуються переваги і недоліки кожної системи [9], а також проводиться порівняння вимог до тренувальних даних і обчислювальних ресурсів на прикладі реальних моделей.

Але системи ШІ та розпізнавання мови можуть відстежувати активність на предмет підозрілої поведінки, надаючи своєчасні сповіщення фахівцям з безпеки та адміністраторам безпеки, таким чином покращуючи можливості виявлення загроз та намірів використання потенційних вразливостей [10, 11].

Розпізнавання мови, інтегроване з DL і технологією блокчейн, може використовуватися для біометричного контролю доступу, забезпечуючи безпеку процесів автентифікації та ідентифікації [12]. Цей підхід допомагає зменшити ризики, пов’язані з безпекою даних, конфіденційністю та витоком інформації. Системи розпізнавання мови можуть виявляти аномалії в поведінці користувачів після надання доступу за біометричними даними (в тому числі, голосу), забезпечуючи тим самим додатковий рівень безпеки. Це особливо корисно для виявлення несанкціонованого доступу або незвичайних дій в системі.

Система ASR, яка описана в [13], вразливі до ворожих атак. Дослідження підкреслює потребу в надійних механізмах захисту, таких як згладжування сигналу і навчання на атаках зловмисника, для захисту систем ASR від таких загроз. Інтеграція розпізнавання мови зі ШІ і ML дозволяє створювати системи в [11], які не тільки розпізнають мову, а й генерують та інтерпретують інформацію, пов’язану з безпекою. Це допомагає представляти критичні висновки про безпеку в зрозумілій для користувачів формі.

Для того, щоб повною мірою використовувати NLT, необхідно поєднати два основних стеки технологій:

– мовленнєві технології для перекладу «мовлення в текст» і навпаки;

– NLP для розуміння, інтерпретації та генерування інформації в тексті.

Розглянемо існуючі знання, напрямки та шляхи майбутніх досліджень у цій дедалі важливішій сфері NLT/NLP, що набуває дедалі більшого значення.

NLP – це область ШІ, яка допомагає комп'ютеру розуміти і генерувати текст. NLP широко використовується в багатьох завданнях: діалогових системах, аналізі настроїв, машинному перекладі, пошуку інформації, узагальненні, відповідях на запитання тощо. За останнє десятиліття відбулося кілька проривів у галузі DL, спочатку для розпізнавання зображень, а потім і для природної мови, що привертає величезний інтерес дослідників і бізнесу. Розглянемо найвідоміші та найфундаментальніші методи, які значно покращують навички машинного розпізнавання природної мови: RNN, концепцію вбудовування, концепцію *декодера* та *енкодера*, а також коротко про *увагу* та *трансформери*. Без цієї техніки важко уявити такий інтерес до NLP.

Основна ідея розпізнавання мови полягає в перетворенні захоплених аудіосигналів у відповідне текстове представлення (рис. 1.1).

Рис. 1.1. Процес розпізнавання природної мови

Система ASR представляє вхідну послідовність звуків (або акустичний вхід) $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ довжини T як результуючу послідовність $\mathbf{Y} = \{y_1, y_2, \dots, y_L\}$ (які можуть бути символами, фрагментами слів або словами) довжини L.

ASR та генерація мовлення – це набір технологій для перетворення людської мови в текст і назад. Після того, як розвинулася структурована система комунікації, яка називається мовою, мова стала основним інструментом будь-якого спілкування між людьми. Для машин такою мовою є цифри. Минуло багато ітерацій, щоб представити людську мову мовою, зрозумілою машині [7].

Незважаючи на те, що гібридні системи демонструють чудові результати, а потреба в навчальних даних і ресурсах все ще менша, ми повинні визнати, що

майбутні роботи в основному зосереджені на онлайнових комплексних системах, які в майбутньому проникнуть у більшість сфер застосування: Internet of Things, голосові асистенти, людино-машинна комунікація тощо. Тому ми повинні приділити їм окрему увагу і детально дослідити їх у наших подальших роботах.

Переглядаючи кілька статей, присвячених онлайн-системам, ми помічаємо, що автори зосереджуються на вирішенні проблем великих вимог до даних, частоти помилок у словах (від англ. Word Error Rate, WER) для мов без якісної вимовної лексики, а також на деяких інших вузьких тактиках для досягнення кращих результатів [14].

Одну з цікавих ідей запропонувала команда дослідників з Facebook та Microsoft. У роботі [15] вони використали стратегію ініціалізації навчання «вчитель-учень» для перенесення знань зі складної офлайнової наскрізної моделі до онлайнової наскрізної моделі розпізнавання мови. І це допомогло їм усунути потребу в якісному лексиконі або будь-якому іншому лінгвістичному доповненні. Ця ідея була оцінена на задачі персонального асистента Microsoft Cortana і показала, що запропонований метод призводить до відносного покращення WER на 19% порівняно з випадково ініціалізованою базовою системою.

Ще більш цікавим для нас виявився напрямок досліджень, спрямований на зменшення вимог до навчальних даних в онлайнових наскрізних системах. Автори роботи [16] зосередили свої дослідження саме на цій ідеї. Важливість цього напрямку пояснюється практичними потребами. Практикам часто доводиться будувати системи штучних нейронних мереж (ШНМ) для нових задач з обмеженими даними про предметну область і в короткі терміни. Хоча нещодавно розроблені наскрізні методи значною мірою спрощують конвеєри моделювання, вони все ще страждають від проблеми з даними. Вони дослідили кілька методів для побудови онлайн-систем ASR наскрізним способом, з невеликою кількістю транскрибованих даних у цільовій області. Ці методи включають доповнення даних у цільовій області, адаптацію області з використанням моделей, попередньо навчених на великій вихідній області (трансферне навчання), і дистиляцію знань на нетранскрибованих даних цільової області з використанням адаптованої

двонаправленої моделі в якості вчителя; вони застосовні в реальних сценаріях з різними типами ресурсів. Ці експерименти продемонстрували, що кожна методика незалежно корисна для покращення продуктивності онлайн-аналізу знань у цільовій предметній області.

1.3. Огляд методів обробки природної мови

Розглянемо основні точки прориву в області NLP за останнє десятиліття. Почнемо з RNN як основної концепції в NLP (повторюваність та об'єднання інформації з попередніх ітерацій), а потім представимо більш просунуті методи інженерії ознак (не просто однозначне кодування слів у наборі даних, а складне векторне представлення з контекстом та додатковою інформацією, пов'язаною зі словом).

Оскільки основною відправною точкою NLP був машинний переклад, природно, що поняття енкодер-декодер і послідовність-послідовність розвивалися, що також буде висвітлено. Увага та трансформер будуть розглянуті як останні досягнення в галузі NLP.

Подальші завдання NLP (машинний переклад, аналіз настрою, відповіді на запитання, виділення частин мови та багато інших) зазвичай вирішуються за допомогою різних підходів, обраних для конкретної задачі. Як правило, йдеться про кероване навчання на наборах даних для конкретної задачі, що є досить трудомістким з точки зору дослідницьких годин і обчислювальних ресурсів. На додаток до цих незручностей, системи, побудовані за таким підходом, дуже чутливі до специфікацій завдань і змін у розподілі даних. Сучасні тенденції рухаються в бік некерованих універсальних моделей і переносу навчання на попередньо навчені моделі з подальшим доопрацюванням. Техніки та компоненти, які пропонуються для огляду, відповідають сучасним тенденціям та революційним досягненням у NLP [17]. У ній окреслено архітектуру сучасних моделей попереднього навчання [18].

Все починається з підготовки даних для навчання або тестування. Вхідні дані проходять через деякі методи, щоб отримати в результаті вбудовування слів [19, 20] або контекстне вбудовування, яке можна описати як багатовимірне вбудовування знань про слово. Незважаючи на використання терміну «слово», не слід плутати зі словом з природної мови. Словесне вбудовування – це форма вектору, який може базуватися на символах, підсловах, словах, реченнях або навіть довших послідовностях, кожна з яких називається токеном. Контекстне вбудовування [21] використовується як вхідні дані для енкодера, який формує вектор контексту і передає його декодеру. Декодер, у свою чергу, формує набір ймовірностей, необхідних для обчислення виходу.

1.3.1. Рекурентні нейронні мережі

RNN були основним будівельним блоком для задач NLP протягом тривалого періоду. Основна відмінність RNN від інших архітектур полягає в їхньої здатності запам’ятовувати дані для послідовності, а не тільки для останньої комірки (слова/лексеми).

Мережа приймає **X** як вхідний вектор (зазвичай закодоване представлення слова) і виробляє **Y** як вихідний вектор. Кожна комірка RNN приймає як поточний вхід x_t, так і попередній прихований стан (активацію) h_{t-1}, в якому зберігається інформація, отримана під час попередніх ітерацій. Мережа навчається вагам (параметрам) W_h, W_x та зміщенню b_a через процес навчання вагів. На кожній ітерації прямого поширення для обчислення вихідного прихованого стану (активації) застосовується нелінійна активаційна функція g, така як *tanh* (або подібна)

$$h_t = g(W_h h_{t-1} + W_x x_t + b_a). \quad (1.1)$$

Крім того, *softmax* (функція активації g) може бути застосована в кінці, якщо вихідні прогнози є необхідними для задачі

$$y_t = g(W_y h_t + b_y). \quad (1.2)$$

Найважливішим для завдань у сфері NLP є те, що вихідні дані включають інформацію з попередніх, а не лише з останніх. Це важливо, головним чином, через природу мови. Одного останнього слова (лексеми) недостатньо для розуміння контексту речення. Такий тип зв'язку називається рекурентним зв'язком (див. рис. 1.2).

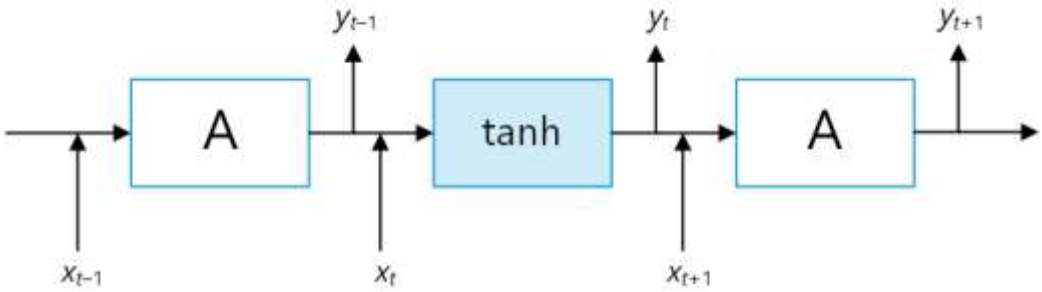


Рис. 1.2. Фрагмент процес функціонування рекурентної нейронної мережі

Ідея та концепція повторюваного зв'язку і контексту суттєво впливають на сучасний стан NLP. RNN мають багато недоліків, таких як односпрямованість, проблема з фіксацією середньо- та довгострокових зв'язків/залежностей всередині послідовності. Сьогодні нечасто можна зустріти RNN в якості базової архітектури. На основі RNN з'явилися складніші архітектури, такі як вентильний рекурентний вузол (від англ. Gated Recurrent Units, GRU), LSTM, а деякі архітектури прийшли в NLP з розпізнавання зображень, наприклад, CNN [22–25].

Для того, щоб нейронна мережа могла виконати своє завдання, необхідно забезпечити числове представлення вхідної послідовності у вигляді токенів. Методи вбудовування слів створюють вектори з лексем. Порівняння векторів призводить до семантичної схожості токенів. Методи вбудовування, такі як GloVe та Word to Vector, пояснюють концепцію моделювання вхідної послідовності через представлення [20].

Основна мета вбудовування – представити токени (документи, фрази, контекст, частину слова або символ) у вигляді числового вектору. Тоді нейронні мережі можуть обчислити і використати розподіл ймовірностей або ймовірності для розділення семантично схожих категорій. Таким чином, різні лексеми зі

схожими значеннями матимуть ближчі вектори, а різні за значенням групи лексем можна буде розділити у векторному просторі. Ферт [18] зробив популярною ідею «слово характеризується компанією, яку воно підтримує». Останнім часом з'явилися нові підходи. Контекстне вбудовування [21] це представлення лексеми в її контексті, що означає отримання інформації про використання лексеми в різних контекстах і кодування знань, які можна переносити на інші мови. Під час вбудовування враховується інформація про присутність лексеми в різних контекстах [26–30].

1.3.2. Високорівнева архітектура енкодерів-декодерів

Підхід енкодер-декодер став проривом і призвів до значного підвищення продуктивності мовних моделей. Вхідна послідовність [«What's»|«up»|«?»] на рівні вбудовування отримує числове представлення, потім числове представлення послідовно подається на RNN. Зрештою, вхідні дані проходять через RNN, що виробляє вихід. Ця частина називається енкодер, яка кодує вхідні послідовності (див. рис. 1.3). RNN обробляє вхідні дані, вбудовуючи їх послідовно (зліва направо), переходячи до наступної часової мітки RNN прихованого стану, обчисленого в поточній часовій мітці RNN. Після того, як всі входи переходять до фінальної часової мітки, фінальна часова мітка RNN виробляє вихід, що представляє всі вхідні послідовності в одному *прихованому стані*. Ця частина називається енкодер, оскільки її основним завданням є не генерація передбачень, а кодування вхідних послідовностей. Після того як енкодер закінчує кодування, прихований стан передається декодеру, завданням якого є декодування та генерація передбачень на основі вхідного прихованого стану. Декодер послідовно приймає на вхід для кожної поточної часової мітки вихідну активацію попередньої часової мітки RNN і вихідне передбачення попередньої часової мітки RNN. Для першої часової мітки він приймає маркер *початку речення* 'Beginning of Sentence Token' як передбачення попереднього шару. Декодер генерує передбачення до тих пір,

поки не згенерує маркер *кінця речення* (від англ. End of Sentence Token, EOS) в залежності від реалізації може бути до певної довжини або іншого параметра.

Результат роботи енкодера передається декодеру. Декодер генерує передбачення отриманої послідовності, поки не дійде до токена кінця речення.

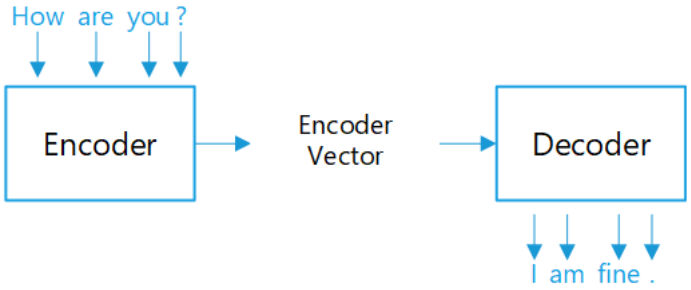


Рис. 1.3. Приклад функціонування енкодера-декодера

Найбільшим досягненням енкодера-декодера є можливість використовувати його для створення наскрізних моделей, а також можливість обробляти вхідні та вихідні послідовності різної довжини. Проблема неузгодженості довжини вхідних і вихідних даних особливо актуальна в нейронному машинному перекладі.

Архітектура енкодера-декодера зазвичай базується на двох RNN або LSTM. Енкодер кодує всі вхідні послідовності і зберігає всю інформацію у векторі енкодера. Декодер створює передбачення результату [31–33].

Основним обмеженням RNN є відстеження залежностей у довгих реченнях. Довгі речення (більше 20 слів) просто не можуть ефективно зберігатися у вихідному векторі RNN. Саме тому дослідники придумали механізм уваги.

Ідея уваги – це те саме, що й увага в процесі читання. Людині достатньо кількох слів у реченні, щоб добре його зрозуміти. Так само і в процесі перекладу: людині потрібно лише кілька основних слів для перекладу, всі інші слова просто випадають з уваги. Так само і з увагою: на кожному кроці декодер фокусується на певній частині джерела. На кожному кроці декодер фокусується лише на окремих словах (більша насиченість означає більшу увагу), а не на всій вхідній послідовності. Механізм уваги надає декодеру таку можливість за допомогою ваг уваги та вектору контексту [34–36].

Трансформер – це один з останніх проривів, який значно прискорює NLP. Архітектура трансформера побудована на основі концепції енкодера-декодера і значною мірою базується на концепції уваги. Основним проривом стало розпаралелювання шляхом повної заміни послідовних обчислень (RNN або CNN) мережею, що базується на увазі. Основні компоненти та концепції цієї архітектури будуть представлені та описані нижче.

Енкодер складається з декількох шарів, що складаються з самоконтролю та подачі із залишковими зв'язками, а також позиційного енкодера. Як завжди, шар вбудовування застосовується внизу для перетворення вхідної послідовності в числове представлення. Мережа прямого поширення не має залежностей і тому може бути розпаралелена. Це важлива концепція, що лежить в основі можливості трансформерів навчатися на дійсно великих обсягах даних, чого не можуть собі дозволити LSTM і GRU.

Декодер також складається з декількох (стільки ж, скільки і енкодер) шарів самоуваги, шарів прямого зв'язку із залишковими зв'язками і додатково шару уваги енкодера-декодера посередині. На відміну від енкодера, шар самоуваги декодера відрізняється. Основна ідея тут – маскування майбутніх позицій. У енкодері кожна позиція може звертати увагу на всі позиції, але у декодері, щоб запобігти лівому потоку інформації і зберегти властивість авторегресії, кожна позиція може звертати увагу лише на ранні позиції у вихідній послідовності [2, 6, 33].

1.3.3. Закритий рекурентний блок і довга короткочасна пам'ять

У відповідь на проблеми середньо- та довгострокової залежності дослідники пропонують дві архітектури, в основі яких лежить ідея залишкового зв'язку 'Cell State'.

Основна ідея GRU [17, 25] полягає у фіксації довгострокових залежностей шляхом додавання комірки пам'яті (C_t), яка в GRU дорівнює прихованому стану (активації), тоді вектор прихованого стану в момент часу t дорівнює

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}, \quad (1.3)$$

де h_{t-1} – вектор прихованого стану в попередній момент часу $t-1$; $\widetilde{h_t}$ – вектор-кандидат прихованого стану в момент часу t; z_t – вектор оновлення, який визначає, скільки інформації з попереднього прихованого стану h_{t-1} і кандидатного прихованого стану $\widetilde{h_t}$ потрібно зберегти.

Формула відображає механізм оновлення GRU, де вектор оновлення z_t визначає, наскільки новий стан $\widetilde{h_t}$ має впливати на остаточний стан h_t, та наскільки зберігається старий стан h_{t-1}. Значення z_t зазвичай визначаються за допомогою сигмоїдної функції активації.

І кожного разу комірка мітки часу розглядає можливість перезапису цієї комірки зі значенням кандидата по формулі що визначає кандидатний прихований стан $\widetilde{h_t}$ в момент часу t

$$\widetilde{h_t} = \tanh(W[r_t * h_{t-1}, x_t]), \quad (1.4)$$

де tanh – гіперболічна тангенс функція активації, яка обмежує значення в діапазоні від –1 до 1; W – це матриця ваг; r_t – вектор скидання ‘Reset Gate’, який модулює вплив попереднього прихованого стану h_{t-1}; x_t – вхідний вектор в момент часу t; $[r_t * h_{t-1}, x_t]$ – конкатенація модифікованого вектору прихованого стану $r_t * h_{t-1}$ і вхідного вектору x_t.

Формула описує, як кандидатний прихований стан $\widetilde{h_t}$ обчислюється шляхом застосування гіперболічної тангенс функції до лінійної комбінації модифікованого попереднього прихованого стану та поточного вхідного вектора. Вектор скидання r_t дозволяє модулювати вплив попереднього прихованого стану, що дозволяє GRU адаптивно забувати або запам’ятовувати інформацію на різних часових кроках.

Цей механізм є частиною внутрішньої роботи GRU, що дозволяє ефективно обробляти послідовні дані і зберігати довгострокові залежності, важливі для задач NLP, часових рядів та інших послідовностей.

За допомогою двох вентилів, що описуються рівняннями (рис. 1.3)

$$\begin{aligned} \textit{Update Gate: } z_t &= \sigma(W_z[h_{t-1}, x_t]), \\ \textit{Reset Gate: } r_t &= \sigma(W_r[h_{t-1}, x_t]). \end{aligned} \quad (1.5)$$

Вентиль оновлення (*Update Gate,* z_t) приймає значення від 0 до 1 (у більшості випадків близьке до 0 або 1), обчислюється шляхом застосування сигмоїдної функції активації до поточної часової мітки на вході x_t та попередньої часової мітки прихованого стану (активації) h_{t-1} з вивченими вагами (параметрами) W_z в процесі навчання ваг. *Update Gate* є основним процесором, що приймає рішення про оновлення прихованого стану, як показано у рівняннях. *Update Gate* вирішує, скільки інформації з попередньої часової мітки слід зберегти на майбутнє. Вентиль очищення (*Reset Gate, r)*, з іншого боку, вирішує, скільки інформації з попередньої часової мітки слід видалити.

Ці ворота оновлення та скидання є ключовими концепціями GRU і вирішують проблеми залежностей базових RNN.

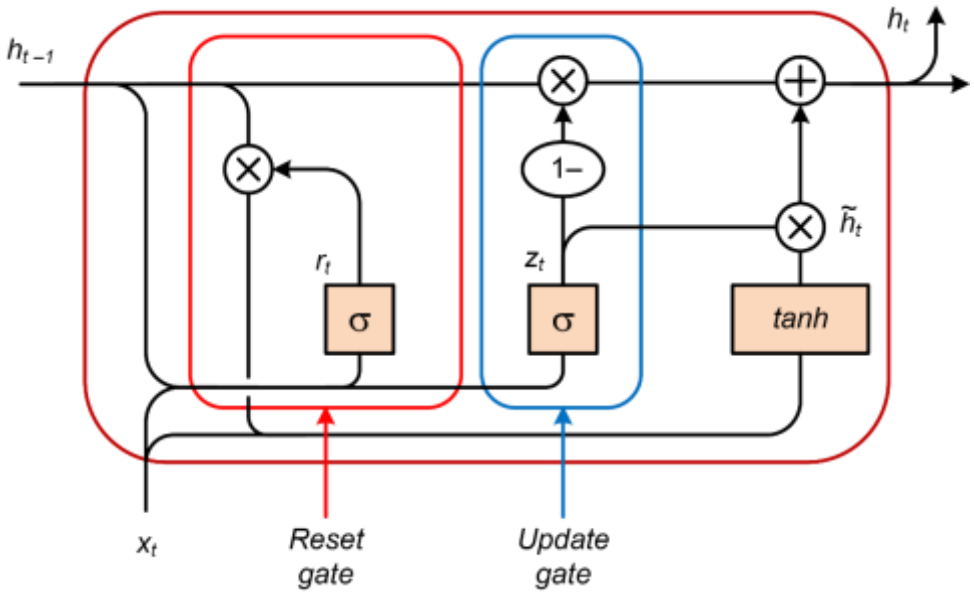


Рис. 1.4. Структура комірки GRU

Інший тип архітектури, який може фіксувати середньострокові залежності, навіть більш потужно, ніж GRU, – це LSTM [37]. На відміну від GRU, яка має два вентилі, LSTM має три вентилі. Важливою концепцією в LSTM є те, що комірка пам'яті (C_t) більше не дорівнює вихідному прихованому стану (активації) h_t. Вихідний прихований стан поточної часової мітки в LSTM переходить до наступної комірки не сам по собі, а з оновленим значенням комірки пам'яті.

LSTM також використовує два окремих вентилі (Gate Update і Gate Forget) для оновлення значення комірки пам’яті, замість того, щоб використовувати один вентиль Update в GRU (який або зберігає, або забуває попереднє значення комірки пам’яті). І замість того, щоб використовувати вентиль скидання в значенні кандидата, він використовує поелементне множення на значення комірки пам’яті, як показано на рис. 1.5.

Вхідний вентиль

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right) \tag{1.6}$$

вирішує, яку важливу інформацію зберігати, а вентиль забуття

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right), \tag{1.7}$$

вирішує, яку і скільки інформації не зберігати, іншими словами, забути (який може перетинатися або не перетинатися з вхідним вентилем). Вхідні та забути вентилі роблять це, використовуючи прихований стан попереднього часового кроку h_{t-1} та поточний вхід x_t. Обидва вентилі використовують сигмоїдну функцію активації, що дає можливість у більшості випадків мати значення вентилів близькими до 0 або 1.

Використання окремих воріт оновлення та воріт забуття для обчислення значення комірки пам’яті

$$C_t = \sigma(f_t * C_{t-1} + i_t * \widetilde{C_t}) \tag{1.8}$$

дає комірці пам’яті можливість не тільки зберігати нову інформацію у комірці пам’яті поточного часового кроку, використовуючи кандидата комірку пам’яті ($\widetilde{C_t}$), але також можливість зберігати деяку кількість інформації з попереднього часового кроку (C_{t-1}). Вихідний вентиль

$$o_t = \sigma(x_t U^0 + h_{t-1} W^0), \tag{1.9}$$

в кінці використовує для обчислення поточного прихованого стану виходу

$$h_t = \tanh(C_t) * o_t, \tag{1.10}$$

на основі оновленого значення комірки пам’яті, обчисленого раніше.

Стан клітини прямолінійний. Він протікає по всьому блоку з незначними лінійними змінами. Ось чому дві запропоновані архітектури дуже добре запам’ятовували довготривалі залежності.

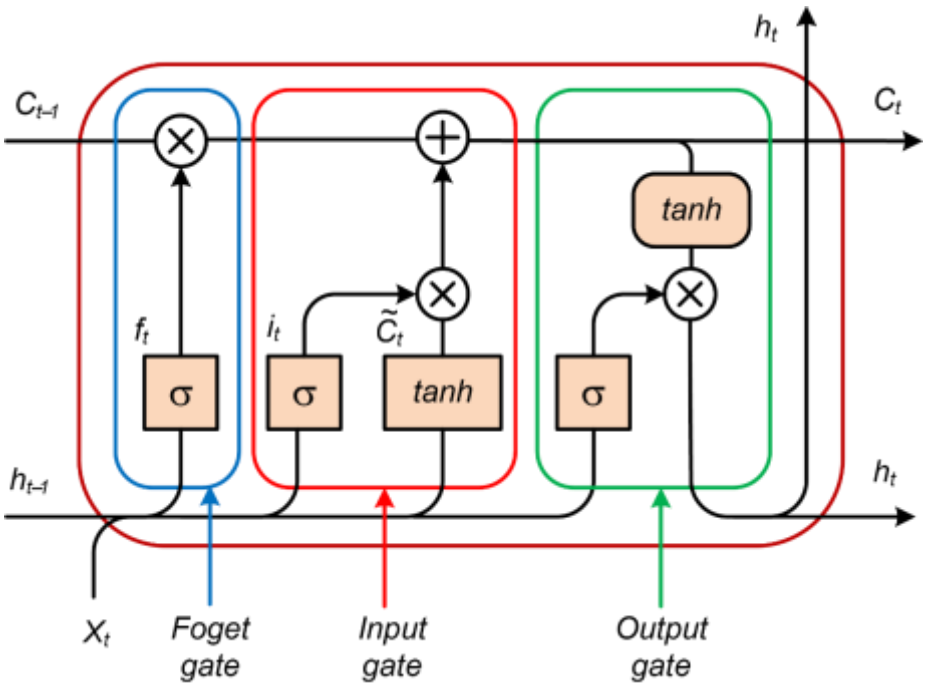


Рис. 1.5. Структура комірки LSTM

Ці мережі є досить споживацькими для обчислювальних ресурсів. Крім того, цей тип архітектури не можна розпаралелити: отже, навчання на великому масиві даних коштує дуже дорого. І незважаючи на те, що він набагато краще працює з довгими послідовностями, на послідовностях, що містять більше 20 слів, спостерігається помітна втрата продуктивності. Ретроспективно застосування RNN, GRU та LSTM стало основним етапом у сучасному NLP, який суттєво вплинув на подальший розвиток галузі.

1.3.4. Двонаправленість у рекурентних нейронних мережах

Крім того, було проведено значну роботу над двонаправленістю ШНМ, щоб надати моделям можливість захоплювати і використовувати інформацію як з більш ранніх, так і з більш пізніх етапів послідовності.

Якщо говорити простими словами, то двонаправлені RNN (від англ. Bidirectional Recurrent Neural Network, BRNN) [37] – це модифікація ШНМ, GRU і LSTM, що складається з двох ШНМ, які одночасно знімають інформацію в протилежних напрямках і лише потім роблять прогнози. BRNN має прямий

рекурентний шар (компонент) S, який приймає на вхід струм **X** і подає на вихід, щоб допомогти передбачити поточний вихід **Y** вперед у часі. З іншого боку, зворотний рекурентний шар (компонент) s_i', який приймає на вході поточний струм **X** і подає на виході інформацію, яка допомагає передбачити поточний вихід **Y** назад у часі.

Для побудови ще потужніших моделей дослідники пропонують складати блоки RNN/LSTM/GRU в стек. Такий тип архітектури називається Deep RNN. Вузьким місцем BRNN є те, що йому потрібна вся послідовність даних перед тим, як робити будь-які прогнози. Deep RNN також набагато дорожчий в обчисленнях. Усі представлені мережі мали проблеми з нейронним машинним перекладом, оскільки вхідні та вихідні послідовності регулярно мали різну довжину через різну мовну семантику.

1.3.5. Підхід для формування уваги

У центрі уваги дослідників була проблема довгих речень (речення містить більше 20 слів), які неможливо ефективно зберігати в одному вихідному векторі RNN/GRU/LSTM. В роботі [34] продемонстровано значне покращення результатів оцінки BLEU за допомогою механізмів уваги. Як вирішення проблеми було запропоновано механізм уваги.

Увага ‘attention’ полягає в тому, що люди читають і запам’ятовують не цілі довгі речення одразу, а по частинах. А для декодера під час декодування (наприклад, перекладу) було б корисно знати, на яку частину вхідної послідовності слід звернути більше уваги. Ідея уваги: на кожному кроці декодер фокусується на якійсь певній частині джерела. На кожному кроці декодер фокусується лише на окремих словах (більша насиченість означає більшу увагу), а не на всій вхідній послідовності.

Механізм уваги відкриває декодеру таку можливість за допомогою ваг уваги та вектору контексту.

На додаток до BRNN, який також може бути двоспрямовані GRU і двоспрямовані LSTM (від англ. Bidirectional LSTM, BLSTM), концепція уваги використовує ідею вирівнювання оцінок і ваг уваги (кількість уваги, яку декодер повинен приділяти при обчисленні прогнозу поточного часового кроку).

Приховані стани енкодера передаються разом з останнім прихованим станом енкодера до декодера. Центральна обробка відбувається у декодері. На кожному часовому кроці декодер обчислює набір ознак (про слова та навколишні слова), які називаються балами вирівнювання – різниця між прихованими станами енкодера та декодера

$$e_{ij} = a(s_{i-1}, h_j), \tag{1.11}$$

які використовуються для обчислення ваги уваги

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{T_x} \alpha_{ij} h_j} \tag{1.12}$$

за допомогою функції *softmax*.

Вектор контексту

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{1.13}$$

обчислюється для кожного кроку декодування шляхом комбінування ваг уваги з попередніми виходами декодера, які передаються на декодер RNN

$$\alpha_{t_s} = \frac{\exp \text{score}\ (h_t, \overline{h_s})}{\sum_{s'=1}^{\mathbf{S}} \exp \text{score}\ (h_t, \overline{h_{s'}})}. \tag{1.14}$$

Хоча дивно, що така проста і типова архітектура, як двонаправлений LSTM з увагою (всього кілька рівнянь і кілька десятків рядків коду) може передбачати (перекладати, класифікувати) з таким чудовим результатом, ця архітектура припускається помилок і має вузькі місця [2]. Цей тип архітектури не може бути розпаралелений – механізм уваги, передбачений для послідовних ШНМ, допоміг вирішити проблеми довготривалих залежностей, використовуючи більш відповідний контекст на кожному кроці, але проблема розпаралелювання обчислень постала ще гостріше.

Крім того, якщо аналізувати сферу NLP не лише через призму перекладу, де більшість машин часу просто перекладають речення за реченням, а зосередитися на розумінні природної мови, то ШНМ не показують хороших результатів у загальному розумінні контексту та моделюванні, особливо під час задач генерації тексту. Це саме та сфера, де архітектура *трансформера* може працювати краще.

1.3.6. Підхід для формування самоуважності і трансформації

У публікації [2] дослідники з Google представили трансформер – нову нейромережеву архітектуру для розуміння мови, засновану на механізмі самоуваги. Високорівнева архітектура енкодера та декодера трансформера представлена на рис. 1.6 і детально описана нижче. Основна новизна полягала в тому, що для побудови мовної моделі взагалі не потрібні ні RNN, ні CNN шари. Достатньо лише шарів самоуваги та прямого поширення.

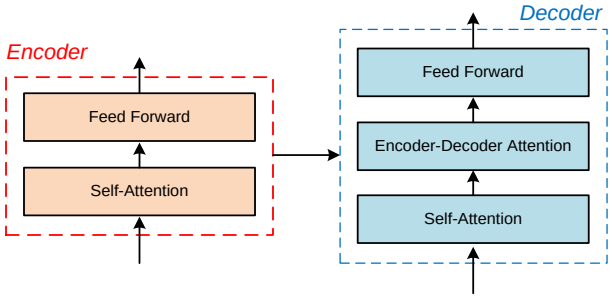


Рис. 1.6. Архітектура енкодера-декодера із трансформером

Трансформер включає в себе багато з того, що було описано раніше: двонаправленість; архітектуру енкодер-декодер для підтримки різної довжини входу-виходу; самоуважність на додаток до уваги, дослідники розвивають ідею уваги і представили самоуважність в трансформері; розпаралелювання (принаймні на кроці Feed-Forward, який є найдорожчим) шляхом повної заміни послідовних обчислень (RNN або на основі згортки) на мережу на основі уваги.

Шари самоуваги в енкодері допомагають моделі зрозуміти, на яких частинах вхідної послідовності (словах) слід зосередитися під час кодування послідовності.

Найважливішим нововведенням є використання трьох векторів: Вектор запиту, Вектор ключа та Вектор значення для створення проекції «запиту», «ключа» та «значення» кожного слова у вхідному реченні. Також іншим важливим рівнем декодера є рівень уваги енкодера-декодера, який отримує вихідні дані останнього рівня уваги енкодера як вхідні дані і використовує вектори уваги Key і Value, щоб сфокусуватись на відповідних місцях вхідної послідовності [22].

1.4. Аналіз методів автоматичного розпізнавання мови

Майже всі сьогодні використовують ту чи іншу техніку вбудовування для токенізації вхідних послідовностей. Якщо ваша задача не є вузькоспецифічною, ви, ймовірно, в кінцевому підсумку використаєте одне з попередньо навчених вбудовувань з розмірністю (300–512). А якщо ви використовуєте специфічні для домену задачі, то вибір буде просто заснований на доступних обчислювальних ресурсах. Починаючи з простих Word2Vec та Glove і переходячи до просунутих методів контекстного вбудовування та збільшеної розмірності, ви зможете вирішити свої завдання з високою точністю.

На сьогоднішній день трансформер є найпотужнішою архітектурою, яку можна навчити на величезних обсягах навчальних даних з мільярдами параметрів. Зрозуміло, що навчати таку велику мережу щоразу з нуля і для кожної конкретної задачі недоцільно (навіть сьогодні це коштуватиме сотні тисяч доларів і величезних обчислювальних потужностей GPU). Тому такі великі моделі постачаються у вигляді попередньо навчених моделей, які потім можуть бути тонко налаштовані для різних сценаріїв і завдань. Це може бути досягнуто за допомогою додаткового шару нейронів на кінці, які не були навчені під час попереднього навчання, і навчити їх як частину нової моделі для конкретних завдань.

Ключовою перевагою моделей, побудованих з використанням трансформерної архітектури, є те, що їх не потрібно навчати на розмічених даних, тому вони можуть навчатися на будь-якому очищеному сирому тексті. Це дає

можливість працювати з дуже великими наборами даних і призводить до ще більшої точності.

Нинішня таблиця лідерів у різних змаганнях з NLP все більше звужується до великих технологічних корпорацій, а не університетів. Це пов’язано з наявністю обчислювальних потужностей. З іншого боку, результати лідерства та практичне застосування відрізняються. Навіть сьогодні багато виробничих сервісів використовують ШНМ (модифікації LSTM та GRU), наприклад, для класифікації намірів у широко розповсюджених фреймворках чат-ботів.

На рис. 1.7 запропоновано алгоритм вибору сучасного підходу до розв’язання бізнес-задач NLP з урахуванням специфікації предметної області задачі та наявності ресурсів. Також доцільно порівняти результати BLSTM з архітектурами на основі уваги та трансформерів з точки зору точності, споживання ресурсів, часу на навчання (а отже, і на вдосконалення), інтерпретованості.

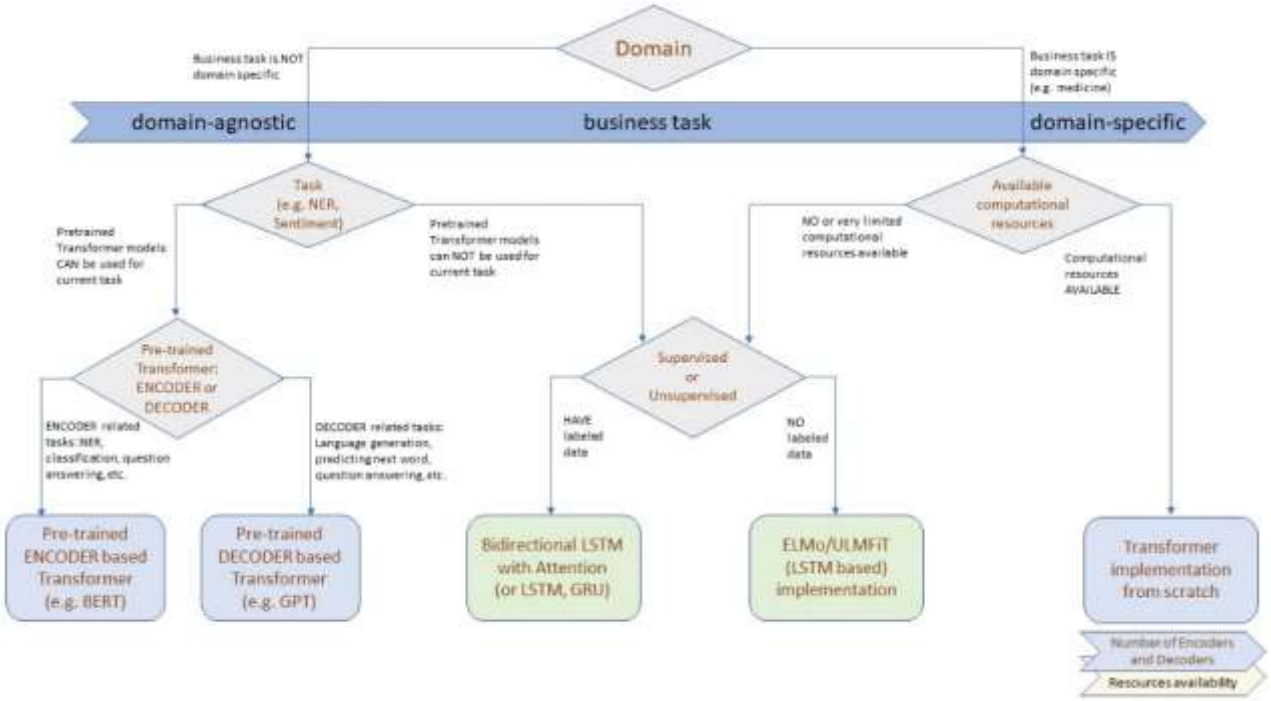


Рис. 1.7. Блок-схема вибору відповідної архітектури NLP для бізнес-задачі

Основною метою мовних систем є перетворення вхідної послідовності звукових хвиль у текстове представлення у випадку системи ASR і навпаки у

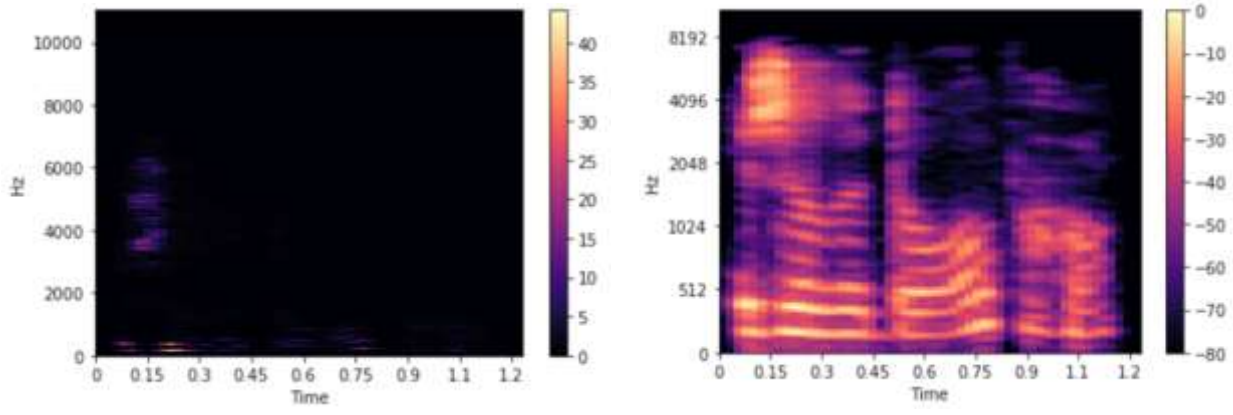
випадку генерації мовлення з тексту (від англ. Text-to-Speech, TTS) [38]. Тут є два основні підходи:

– гібридні моделі на базі систем ASR;
– комплексні системи ASR.

1.4.1. Прихована марковська модель

Для того, щоб представити вхідну аудіопослідовність у форматі, зрозумілому машині, ми повинні здійснити деякі перетворення. Як показують дослідження, недостатньо просто перетворити вхідну хвилю в цифри відповідних амплітуд шляхом дискретизації аудіосигналу. Такі функції просто дуже неінформативні для процесу навчання, щоб витиснути якомога більше інформації для запам'ятовування та узагальнення аудіосигналу.

Спектрограму було отримано за допомогою швидкого перетворення Фур'є (ШПФ) для представлення часу (або подібних нелінійних характеристик), частоти та енергії в кожній точці часу. Перетворення представляє особливості у форматі акустичних кадрів (20–40 мс). Мелчастотні кепстральні коефіцієнти (від англ. Mel-Frequency Cepstral Coefficients, MFCC) або перцептивне лінійне передбачення є загальним вибором методів нелінійного перетворення для виділення ознак для даних ASR [39, 40]. Класичне ШПФ не підходить для визначення закономірностей (див. рис. 1.8).



а б

Рис. 1.8. Приклади спектрограм: (а) ШПФ і (б) MFCC

Щоб відновити висловлювання, яке щойно прозвучало, поставивши правильні фонеми одна за одною, було використано приховану марковську модель (від англ. Hidden Markov Model, HMM). Це робиться за допомогою статистичних ймовірностей того, що одна фонема слідує за іншою (рис. 1.9). Моделювання систем ШІ полягає у створенні генеративної моделі [41]. Класичним способом створення ASR є побудова моделі мови, моделі вимови та акустичної моделі. До недавнього часу всі три компоненти були необхідними для вирішення задачі розпізнавання мови.

Рис. 1.9. Взаємозв’язок елементів багатомодульної системи автоматичного розпізнавання мови

Спрощено кажучи, HMM складається з трьох різних шарів:

1. Серцем моделі HMM є акустична модель, яка перевіряє на акустичному рівні ймовірність того, що фонема, яку вона розпізнала, є саме цією фонемою.

2. Після цього застосовується лексична (вимовна) модель, яка перевіряє ймовірність того, що розпізнані фонеми можуть стояти поруч одна з одною.

3. Зрештою, застосована мовна модель (зазвичай у вигляді *n*-грам) перевіряє на рівні слів, чи мають сенс слова, що стоять поруч, один з одним. Як приклад, модель вибере «cat paws ʻкотячі лапи’» замість «cat pause ʻкотяча пауза’» [42–46].

З мовної моделі створюємо послідовність слів. Модель вимови дає результат того, як вимовляється конкретне слово. Ми бачимо, що воно записане у вигляді послідовності фонем, які є основними одиницями звуку (послідовність токенів). Модель вимови перетворює послідовність тексту на послідовність токенів вимови.

Після цього вона подається в акустичну модель, яка видає результат того, як звучить той чи інший токен.

У цьому конвеєрі кожен компонент має свою статистичну модель. Кінцевим результатом спільної роботи всіх моделей є виведення найбільш ймовірної текстової послідовності $\mathbf{Y}' = \{y_1, y_2, \dots, y_L\}$ за заданими даними (аудіо ознаками) $\mathbf{X}' = \{x_1, x_2, \dots, x_T\}$

$$\mathbf{Y}' = \arg\max_{\mathbf{Y}} p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y}). \quad (1.15)$$

Однак впровадження наскрізних моделей змінило архітектуру, тому немає потреби в багатьох окремих компонентах.

Розглянемо систему розпізнавання мови, які базуються на гібридній HMM-DNN (від англ. Deep Neural Network) архітектурі та наскрізних моделях, приділяючи увагу досягнутим результатам у точності розпізнавання мови. Зокрема, ми проаналізуємо парадигму моделі коннекціоністської часової класифікації (від англ. Connectionist Temporal Classification, CTC), модель Listen, Attend, and Spell (LAS), яка є послідовно-послідовною моделлю, а також найновіші архітектури послідовно-послідовних онлайн моделей. Ми окреслимо їхні переваги та недоліки. На жаль, через велику різноманітність підходів до навчання, архітектур моделей, мов, що використовуються, а також різноманітність навчальних і тестових даних, немає можливості зібрати об’єктивне порівняння результатів навіть з великої кількості робіт, опублікованих у цій галузі. Тому ми наведемо приклади найкращих на сьогоднішній день опублікованих результатів, наскільки відомо авторам, для різних підходів і типів систем ASR.

Як видно з вищесказаного, одним з головних обмежень HMM-моделей є відображення фонем на графеми. Особливо ця проблема постає для мов з низьким рівнем ресурсів, де ніхто не намагався підготувати таку модель. Підготовка набору даних для такого відображення може зайняти багато часу. Це одна з причин, чому з’явилися спрощені наскрізні моделі. Основним натхненням для цього було навчання моделі з якомога меншою кількістю маркувань та проміжних кроків. Модель повинна навчитися самостійно зіставляти фонему з графемою прямим або непрямим способом, використовуючи ті самі вхідні дані, що й для поточного

навчання. Інша мотивація полягає в тому, щоб зануритися в область навчання без нагляду, щоб використовувати величезну кількість нерозмічених аудіоданих, що зберігаються в інтернеті.

Існує небагато різновидів архітектур наскрізних систем ASR, в той же час, всі вони побудовані на двох типах: CTC та послідовність до послідовності (на основі енкодера-декодера).

До появи CTC основним обмеженням наскрізної системи ASR було те, що для початку перекладу модель повинна мати повне речення. Це означає відсутність можливості потокового декодування. CTC відображає вхідну послідовність **X** (MFCC) у вихідну послідовність **Y** (літери).

Одним з проривів CTC є введення локальної уваги, яка розбиває безперервне мовлення, а потім поточний блок моделювання, що використовує увагу, працює над кожним розділеним сегментом. Таким чином, все висловлювання розбивається на невеликі сегменти, і локальна увага використовується для прогнозування ознак (літер) [15, 16, 47–49].

Звичайні системи TTS складаються з декількох частин:

– модифікації RNN (LSTM, GRU тощо) для рекурентної мережі передбачення ознак від послідовності до послідовності, яка відображає вбудовування символів у спектрограми MFCC (опис цих компонентів у загальних рисах дуже схожий на описані вище компоненти);

– система вокодерів, яка синтезує форми хвиль з цих спектрограм. Взаємозв’язок між лінгвістичними ознаками та параметрами вокодера, які представляють характеристики голосових зв’язок і голосового тракту, вивчені на акустичних моделях. Параметри вокодера генеруються (на етапі синтезу) на основі навчених акустичних моделей, а форма мовного сигналу синтезується за допомогою високоякісних вокодерних систем [50–53].

1.4.2. Гібридна прихована марковська модель та нейронні мережі

HMM підходить для моделювання послідовностей спектральних векторів, що змінюються в часі, як дуже ефективний фреймворк [54]. Тому більшість сучасних систем безперервного розпізнавання мови базуються на HMM. Більшість перших систем розпізнавання використовували HMM для моделювання стану мовлення та модель з гауссовою сумішшю викидів (від англ. Gaussian Mixture Emissions, GMM) для моделювання ймовірності спостереження станів HMM. Це вважалося проривом у підходах до розпізнавання мови, аж поки не з'явилися нейронні мережі.

У 2011 році Microsoft Research представила гібридну систему (CD-DNN-HMM), в якій HMM було поєднано з контекстно-орієнтованим DNN [44]. Результат виявився значно кращим порівняно з системою HMM-GMM. У 2012 році HMM-DNN значно перевершила найсучасніші системи HMM-GMM [38].

Загалом, системи розпізнавання мови на основі HMM складаються з трьох частин: акустичної, вимовної та мовної моделей. Кожна з цих частин може бути побудована на HMM у поєднанні з нейронними мережами. Маючи свою статистичну модель, кожен компонент надає свої гіпотези щодо результатів роботи. Таким чином, разом це дає наступне твердження:

$$\arg\max_{\mathbf{Y}} \mathbf{Y} * p(\mathbf{Y}|\mathbf{X}) = \arg\max_{\mathbf{Y}} \mathbf{Y} * \sum_{\mathbf{S}} \mathbf{S} p(\mathbf{X}|\mathbf{Y}) p(\mathbf{S}|\mathbf{Y}) p(\mathbf{Y}), \tag{1.16}$$

де $\mathbf{S}$ – послідовність станів HMM $\mathbf{S} = \{s_t \in \{1, 2, \dots, I\} \mid t = \{1, 2, \dots, T\}\}$, а $p(\mathbf{X}|\mathbf{S})$, $p(\mathbf{S}|\mathbf{Y})$ та $p(\mathbf{Y})$ відображають акустичну модель, модель вимови та модель мови [47].

Незважаючи на високу продуктивність наскрізних систем, гібридні системи HMM-DNN і HMM-(B)LSTM (двонаправлена довга короткочасна пам'ять) домінують у багатьох виробничих середовищах. Реальні проблеми мають свої вимоги, наприклад, у переважній більшості випадків текстових даних значно більше, ніж аудіо, або реальна задача вимагає декількох окремих мовних моделей, тому в цих ситуаціях HMM-DNN/(B)LSTM є логічним вибором [55].

Цікаве порівняння наскрізних та гібридних HMM-DNN було проведено в роботі [56]. Для порівняння гібридних HMM-DNN та наскрізних систем були обрані наступні архітектури. Гібридні DNN/HMM та системи, що базуються на увазі, мали BLSTM для акустичного моделювання/кодування. Крім того, для мовних моделей використовувалися LSTM і трансформерні архітектури. Наскрізна система мала дизайн енкодера-декодера, що базується на увазі. Для навчального набору LibriSpeech найкращий WER, досягнутий для обох систем, становив 8,4% для наскрізної системи та 4,5% для гібридної HMM-BLSTM з трансформерною мовною моделлю. Описана в [56] гібридна система перевершила наскрізну систему на 40%. Ці результати були досягнуті на 960 год. навчальних даних. Але автори також стверджують, що при значному збільшенні обсягу навчальних даних розрив у результатах обох систем різко скорочується. Чим менший обсяг даних, тим ефективнішими є гібридні системи.

Все вищезазначене дає нам підстави зробити висновок, що гібридним системам притаманні наступні обмеження:

– багатомодульна архітектура робить його складним у розробці, навчанні та оптимізації;

– оскільки кожен з компонентів гібридної системи має власну статистичну модель і власну нейронну мережу, вони виробляють незалежні помилки, які не узгоджуються між собою, що робить досягнення високих результатів ще більш складним завданням;

– найбільш впливовим недоліком є те, що застосування нейронних мереж обмежується функцією спостереження станів HMM, що дає нам уявлення про те, що система стикається зі стелею в розвитку.

Однак, як було зазначено вище, гібридні HMM-DNN/(B)LSTM системи все ще мають цінні якості;

– найсучасніші результати, які перевершують комплексні системи у виконанні різноманітних завдань;

– набагато кращі результати на значно меншому обсязі навчальних даних;

– можливість ефективно вирішувати кілька реальних завдань з меншими навчальними та обчислювальними ресурсами.

1.4.3. Наскрізне автоматичне розпізнавання мови

Всі обмеження вищезгаданих гібридних багатомодульних систем стали натхненням для дослідників створити процес, коли вся модель навчається як одна велика модель, яка згодом отримала класифікацію наскрізної моделі. Вона просто перетравлює дані або ознаки $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ і виробляє результуючу послідовність $\mathbf{Y} = \{y_1, y_2, \dots, y_L\}$ за допомогою лише однієї потужної імовірнісної моделі $\mathbf{Y} = p(\mathbf{Y}|\mathbf{X})$.

Перша така модель називається CTC і була представлена в роботі [57]. Вона замінила HMM в архітектурі моделей. Моделі на основі CTC можуть безпосередньо виводити кінцеві транскрипти, тоді як моделі на основі HMM здебільшого виводять невеликі одиниці, такі як фонеми та інші, і для отримання результатів потрібно багато подальшої обробки. Застосування CTC значно спрощує архітектуру та навчання моделі. Свого часу це було великим досягненням. Маючи значні обмеження, він спричинив подальшу ідею або побудову послідовно-послідовних моделей [58, 59]. Однак моделі на основі CTC все ще мають своє місце у виробничому середовищі, як Google, Baidu тощо, тому це важливий момент у розвитку ASR та поточного технологічного стеку.

1.4.4. Коннекціоністська модель часової класифікації

Розпізнавання мови за допомогою моделі на основі CTC проходить через два важливих процеси: обчислення ймовірності шляху та агрегування шляхів. Обчислення ймовірності шляху відбувається наступним чином. Спектрограма (ознаки **X**) подається на двонаправлену RNN. Словником для CTC є мітки, це можуть бути літери $\{a, b, c, \dots, z\}$ і додатковий токен $< b >$, який називається

«порожній токен». Кожен кадр прогнозу – це історія ймовірності для різних класів токенів відповідно до часового кроку. Це називається оцінкою s. А повне рівняння має вигляд:

$$s(k,t) = \log P_r(k,t|\mathbf{X}), \tag{1.17}$$

де *softmax* на кроці t дає оцінку $s(k,t)$*, яка* є логарифмом ймовірності категорії k на кроці t, враховуючи дані $\mathbf{X}$. На кожному кроці RNN видає декілька результатів функції *softmax*. Результатом роботи моделі має бути ймовірність транскрипту через ці окремі ймовірності в часі. Таким чином, система може пройти шлях через весь простір результатів функції *softmax* і подивитися лише на символи, які відповідають кожному з часових кроків.

Потім відбувається агрегація. З процесу обчислення ймовірності шляху можна з'ясувати, що довжина вихідного шляху дорівнює довжині вхідної мовної послідовності, що не збігається з реальними даними. У більшості випадків довжина транскрипції менша за довжину вхідної мовної послідовності. Відображення «багато до одного», «довгий до короткого» – це необхідність об'єднати декілька шляхів у коротшу послідовність міток.

1.4.5. Послідовна модель

Хоча модель CTC є чудовою, з точки зору моделювання, ви побачите, що вона робить прогнози лише на основі поточних даних. І після того, як вона зробить ці прогнози для кожного кадру, немає ніякого способу скоригувати ці прогнози. Вона повинна робити все, що може з цими прогнозами.

Альтернативною наскрізною архітектурою, яка не потребує проміжних кроків, є модель sequence-to-sequence [24], основною функцією якої є генерація прогнозу наступного кроку в будь-який довільний момент часу, використовуючи всі попередні дані. Основним обмеженням застосування послідовності до послідовності для розпізнавання мови є можливість відстеження довгих послідовних залежностей, і якщо в тексті мова йде про 10–20 часових кроків, то

аудіопослідовність, навпаки, набагато довша, і залежності доводиться відстежувати на відстані близько сотні часових кроків. Це обмеження було частково вирішено за допомогою механізму вектору уваги та механізму ієрархічного енкодера, який шукає в дуже вузькому інтервалі навколо поточної часової мітки, щоб побудувати вектор уваги.

Наприклад, LAS є однією з реалізацій послідовного перетворення звукової послідовності. LAS видає кілька виходів з ймовірностями для вхідної послідовності (мультимодальні виходи). Саме тому ця модель може навчатися таким складним функціям, адже чим більше помилкових виходів вона видає, тим більше зворотного зв'язку вона має. Більше того, модель може вивчати дуже специфічні шаблони набору даних, що робить цю модель гарним кандидатом для тонкого налаштування предметної області. Ще однією сильною стороною LAS є причинно-наслідкові зв'язки, що означає, що модель може передбачити, наприклад, цифру замість слова. LAS все ще може отримати вигоду від зовнішньої мовної моделі – вона не замінить мільярди текстів слів для мовної моделі, тому все ж таки краще мати два набори даних, один для ASR моделі, а інший для навчання мовної моделі, щоб використовувати як додатковий шар поверх ASR моделі.

У моделях послідовності для мовлення є кілька обмежень:

– оскільки це модель на основі енкодера-декодера, вона не є онлайн-моделлю, а це означає, що це передбачення наступного токена, і ми повинні мати повну послідовність для отримання результату, ми не можемо давати його по частинах, як у гібридних моделях;

– як наслідок попереднього, ми не можемо генерувати точні початок і кінець слів, використовуючи модель «послідовність до послідовності»;

– механізми уваги є вузьким місцем в обчисленнях аудіопослідовності;

– точність набагато нижча для коротких послідовностей, що робить поточну архітектуру важко реалізованою для мовних діалогових систем, де клієнти можуть вимовляти просто «так» у відповідь на запитання системи [8].

1.5. Порівняльний аналіз мовних моделей та фреймворків

Існує кілька основних поділів підходів, що використовуються для вирішення завдань NLP. Основними з них є або використання попередньо навчених моделей, таких як (BERT [60], RoBERTa [61]), або навчити модель з нуля на основі архітектури BRNN, LSTM, CNN (див. рис. 1.10).

Моделі з попереднім навчанням можна розділити на моделі на основі енкодерів і моделі на основі декодерів. Ми не розглядатимемо попередньо навчені моделі без трансформерів, оскільки з дистильованими попередньо навченими моделями на основі трансформерів можна досягти майже такої ж ефективності моделі, як і BRNN, але зі значно вищою точністю.

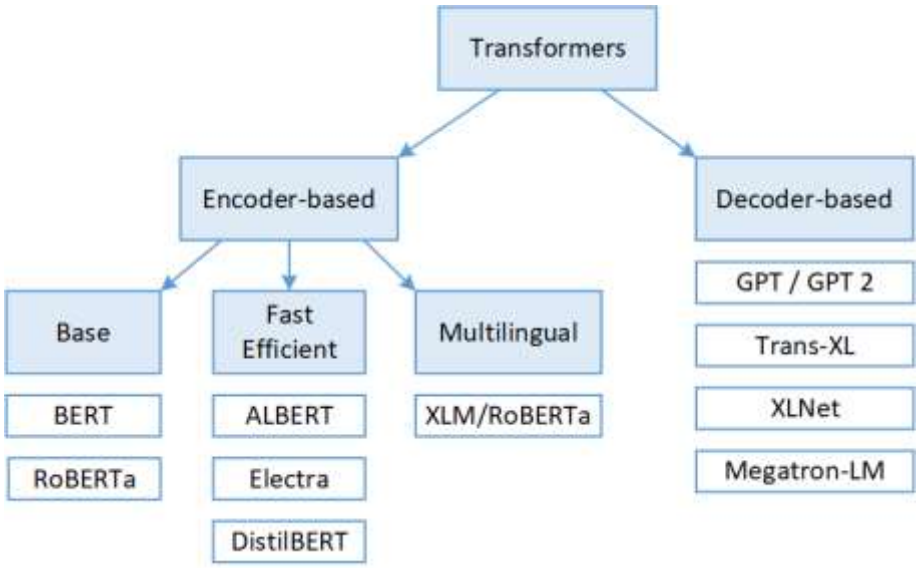


Рис. 1.10. Класифікація моделей [62–69]

Якщо завдання або область специфічні, і ви не можете використовувати попередньо навчену модель, доцільно навчити BRNN, LSTM модель з нуля (див. рис. 1.11).

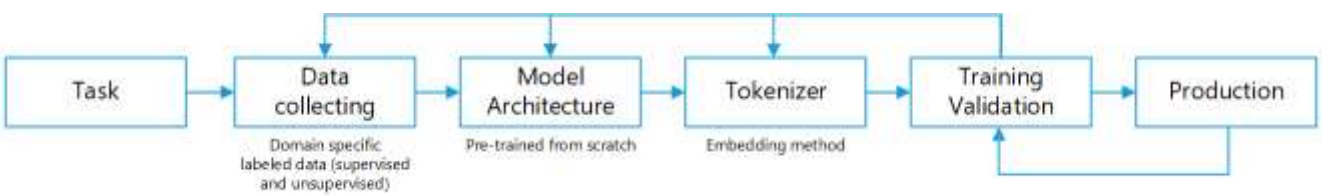


Рис. 1.11. Схема процесу навчання LSTM моделі

Порівняльний аналіз звичайного інструментарію для обробки мовлення (від англ. Speech Processing Toolkit, SPT) та інструментів для побудови моделі ASR представлено в табл. 1.2. Ми порівнювали за кількома параметрами, найважливішими з яких є потреба в обсязі даних для тренування або точного налаштування моделі, а також крива навчання для початку роботи з інструментарієм.

Таблиця 1.2

Порівняльний аналіз існуючого програмного забезпечення

Продукт	Тип SPT	Тип тренінгу	Крива навчання
Kaldi [70]	Гібридна	Під наглядом	Складно
Julius [71]	Гібридна	Під наглядом	Складно
DeepSpeech [72]	Наскрізна	Під наглядом	Легко
EspNet [73]	Наскрізна	Під наглядом	Легко
FairSeq [74, 75]	Наскрізна	Без нагляду	Нормально

Як видно з порівняльного аналізу, жоден інструментарій не може впоратися з усіма етапами попередньої обробки даних ASR (включаючи збір, розбиття, маркування, підготовку до очікуваного формату ASR) та навчання моделей.

З кількості фреймворків та інструментів видно, що досить багато команд розробників займаються в даній галузі, щоб знизити криву навчання та дефіцит в навчальних даних.

1.6. Формування підходів до навчання мовних моделей

Завдяки згаданому прориву в NLP та мовленнєвій сфері все більше і більше компаній бачать великі можливості для впровадження систем на основі NLT. Це зумовлює попит на NLP/мовних інженерів. Такий попит не може бути задоволений існуючою пропозицією. Ось чому крива навчання повинна бути знижена. І одним з основних кроків до збільшення такої пропозиції є наскрізні системи, що набагато простіше з точки зору кінцевого користувача (інженера). Ми бачимо величезну тенденцію і можливість у наскрізних підходах до навчання.

Маркування даних є найбільш складним, тривалим і дорогим процесом – тенденції останнього року полягають у використанні нерозмічених даних. Це набагато простіше, ніж збирати розмічені дані спеціально для вашого домену або мови з обмеженими ресурсами. Однак навіть нерозмічені дані повинні відповідати вимогам. Сьогодні підходи до навчання NLP без нагляду стали стандартними, а з підходами до дистиляції та відсікання на додачу до базового навчання вони стають ефективними та практичними. Ми все ще з нетерпінням чекаємо на більш ефективне неконтрольоване навчання в мовленнєвій сфері, яке потребуватиме не 50 000 год. мовлення та величезних ресурсів для навчання моделі, а буде більш практичним.

Щоб навчити модель узагальнювати нерозмічені дані, потрібно набагато більше даних, а отже, дуже багато обчислювальних ресурсів (50 000 год. нерозмічених даних для навчання найсучаснішої моделі ASR). Саме тому третій тренд – це попередньо навчені моделі. Для NLP зараз поширеним є одноразове навчання моделі на великій кількості даних, а потім тонка настройка для основних завдань. Як приклад можна навести використання попередньо навченої BERT-моделі для розпізнавання розділових знаків після ASR для NLP-обробки. Не кажучи вже про те, наскільки це зменшує час навчання та інвестиції для інженера, щоб підготувати готову до використання модель, оскільки вам не потрібно збирати величезну кількість наборів даних лише для навчання, наприклад, токенізатора. Що стосується мовлення, то попереднє навчання є поширеною причиною для гібридних моделей і все ще потребує розвитку для наскрізних підходів (особливо для неконтрольованого навчання), оскільки зрозуміло, що дослідник або інженер-програміст не матиме можливості витратити сотні тисяч доларів на навчання моделі з використанням неконтрольованих даних.

Оскільки люди вивчають багато мов, щоб розуміти інших людей (навіть у межах однієї країни), то розвиток багатомовних моделей йде в сторону розуміння багатомовних діалогів.

Ми вважаємо, що всі ці тенденції стали можливими завдяки тому, що з’явилися великі спеціалізовані фреймворки для NLP та мовлення, як кілька

прикладів: HuggingFace [76] для NLP, Kaldi [70], ESPnet [73], FairSeq [75] для розпізнавання мови в текст, Tacotron [49] для синтезу TTS. Фреймворки та інструментарій будуть продовжувати розвиватися, збираючи останні досягнення та готуючи інтерфейси для їх використання все більшою кількістю інженерів.

1.7. Постановка наукового завдання дослідження

Беручи до уваги розглянуті вище дослідження та дані про суттєве зростання галузі ШІ та розпізнавання і генерації голосу та природної мови, слід визнати, що зростає спектр і кількість можливих загроз, пов’язаних з обробкою, розпізнаванням та генерацією голосової інформації і тексту, що підкреслює необхідність розробки та впровадження більш ефективних і безпечних методів у цій сфері. Крім того, крім зловмисників, також звичайні користувачі можуть становити певну загрозу через можливість допущення ними помилок або ненавмисних дій, що можуть призвести до порушення інформаційної безпеки [77]. Але при роботі зловмисника рівень збитків є набагато більшим, бо він проводить свої дії точково та по заздалегідь спланованій схемі. Інструментарій зловмисника складається із нелегітимної авторизації, підміни профілю, діпфейків, фішингу, але не обмежений даним переліком. Для навчання своїх моделей зловмисник не також обмежений у виборі інструментів та джерел аудіоданих.

В результаті для протидії зловмисникам потрібно мати в наявності досить точно навчені мовні моделі, щоб знизити ймовірність неправильного розпізнавання суб’єктів, а також для ефективного використання обчислювальних ресурсів. Однак для навчання більш точних моделей розпізнавання, потрібні якісні розмічені дані з постійно включеними векторами атак, до яких мають доступ зловмисники. Для виправлення даної ситуації, підприємства і державні організації повинні впровадити механізми роботи з несегментованими і нерозміченими даними, які не обмеженні вищевказаними критеріями.

Саме тому, розробка політики безпеки має ґрунтуватись на точному розпізнаванні мови і наміру суб’єкта. Таким чином, переважна більшість досліджень зосереджується на розгляді точності роботи із одномовними даними. Однак, у зв’язку із тим що навіть в середині однієї місцевої або міжнародної організації використовується кілька мов, недостатність уваги до проблем точності в розпізнавання природної мови може призводити до великої кількості нелегітимних спрацювань і до неможливості аналізу даних у вбудованих системах і на подальших кроках, виникає необхідність у застосуванні концепцій і методів, які б нівелювали дані обмеження.

Одним зі шляхів вирішення вищезазначеної проблематики є мінімізація ризиків за допомогою навчання користувачів і адміністраторів систем, для ідентифікації загроз через аудіальні канали зв’язку. В порівнянні з такими принципами захисту на основі превентивного навчання, ШІ пропонує ряд рішень, які значно знижують ризик виникнення загроз і намірів, а також мінімізують загрозу поширення навіть успішної поточної атаки шляхом інтеграції в інші системи.

Однак, проведений в попередньому підрозділі аналіз літературних джерел щодо застосування основних концептуальних принципів забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації, свідчить про зосередження уваги здебільшого саме на швидкості розробки нового функціоналу, лишаючи поза увагою етичні питання. Юридичне забезпечення процесів роботи із мовними моделями ще знаходиться на етапі активної розробки, залишаючи відкритим питання щодо меж використання персональної акустичної інформації при навчанні нових та донавчанні існуючих систем.

Таким чином, можна зробити висновок, що на практиці застосування концептуальних принципів безпечного розпізнавання обробки голосової інформації при формуванні мовних моделей загострилося протиріччя між необхідністю активного розвитку технологій навчання, які для збільшення точності потребують нових даних на забезпечення інформаційної безпеки та захисту персональних акустичних даних окремих суб’єктів. Слід зазначити, що

зловмисники не мають юридичних та етичних обмежень на використання будь-яких даних, в тому числі конфіденційних, для навчання власних моделей та використання їх в протиправній діяльності.

У зв’язку з цим, існує необхідність вирішення актуального наукового завдання, сутність якого полягає в подальшому розвитку методів вдосконалення безпечного розпізнавання та параметризації результатів обробки голосової інформації за допомогою вдосконалення збору і маркування легальної аудіоінформації, а також навчання та донавчання мовних моделей, зокрема програмних аспектів їх забезпечення.

Метою дисертаційного дослідження є підвищення ефективності застосування безпечного розпізнавання та параметризації результатів обробки голосової інформації в ІКС завдяки комбінуванню підходів при формуванні розмічених аудіоданих для навчання мовних моделей та в процесі навчання та донавчання цих моделей.

У відповідності до сформованої мети для вирішення зазначеної науково-прикладної проблематики в роботі сформульовані часткові завдання:

– проаналізувати поточний стан і підходи до забезпечення безпеки голосової інформації, як одного із ключових елементів персональних даних суб’єкта, а також розглянути сучасні архітектуру та структуру елементів ІКС, які працюють із аудіоданими;

– провести детальний аналіз метрик природної мови та критеріїв для оцінювання якості її обробки;

– розробити модель автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів та визначити критерії оцінки роботи цієї моделі;

– визначити способи підвищення ефективності розпізнавання мовної інформації при одночасній роботі із кількома мовами при визначенні емоційного стану суб’єкта;

– формалізувати переваги, обмеження, ризики та виклики при впровадженні та застосуванні методів розпізнавання голосової інформації;

– сформулювати вимоги до даних для навчання мовних моделей та дослідити доступні мовні корпуси для української мови;

– покращити сегментацію неформатованого тексту з використанням мовного моделювання та маркування послідовностей;

– дослідити нові підходи до розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, а також запропонувати спосіб підготовки та валідації вхідних даних;

– запропонувати підходи до підвищення точності розпізнавання природної мови для близькоспоріднених мов;

– вибрати мови із низької точністю для проведення експериментів та провести за допомогою них тренінг моделі, а також верифікувати результати експериментів.

Висновки до розділу 1

1. Визначено роль та проаналізований поточний стан і підходи до розвитку та впровадження ML та DL при обробці голосової інформації, як одного із ключових елементів персональних даних суб'єкта, а також розглянуто сучасні проблеми та виклики при використанні нейронних мереж, які працюють із аудіоданими. Встановлено, що ефективна реалізація розпізнавання аудіоінформації в режимі реального часу сприяє підвищенню загального рівня захищеності інформаційних ресурсів підприємства.

2. Проведено аналіз основних підходів, методів та сучасних практик роботи з природною мовою, трансформації машинного навчання в глибоке навчання, еволюцію точності та складності вирішення задач. Також були розглянуті рекурентні нейронні мережі, архітектури та підходи для автоматичного розпізнавання природної мови, а також був проведений порівняльний аналіз мовних моделей та фреймворків. Як показав аналіз, багато з новітніх підходів є занадто вимогливими до обчислювальних ресурсів та алгоритмів для вибору правильної моделі для задачі забезпечення інформаційної безпеки.

3. Сформульовано актуальне наукове завдання, яке полягає в подальшому розвитку методів та засобів забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації для формування систем моніторингу та реагування на інциденти інформаційної безпеки.

Список використаних джерел у розділі 1

1. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. In *Psychological Review* (Vol. 65, no. 6, pp. 386–408). American Psychological Association (APA). https://doi.org/10.1037/h0042519

2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6000–6010).

3. Issaiy, M., Ghanaati, H., Kolahi, S., Shakiba, M., Jalali, A. H., Zarei, D., Kazemian, S., Avanaki, M. A., & Firouznia, K. (2024). Methodological insights into ChatGPT's Screening Performance in Systematic Reviews. In *BMC Medical Research Methodology* (Vol. 24, no. 1). Springer Science and Business Media LLC. https://doi.org/10.1186/s12874-024-02203-8

4. Kraidia, I., Ghenai, A., & Belhaouari, S. B. (2024). Defense against adversarial attacks: robust and efficient compressed optimized neural networks. In Scientific Reports (Vol. 14, no. 1). Springer Science and Business Media LLC. https://doi.org/10.1038/s41598-024-56259-z

5. Wang, F.-Y., Zhang, J. J., Zheng, X., Wang, X., Yuan, Y., Dai, X., Zhang, J., & Yang, L. (2016). Where Does AlphaGo Go: From Church-Turing Thesis to AlphaGo Thesis and Beyond. In *IEEE/CAA Journal of Automatica Sinica* (Vol. 3, no. 2, pp. 113–120). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/jas.2016.7471613

6. Iosifov, I. Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2021). Natural Language Technology to Ensure the Safety of Speech Information. In

Proceedings of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems II (Vol. 3187, no. 1, pp. 216–226).

7. 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 (PIC S&T)* (pp. 335–337). https://doi.org/10.1109/picst51311.2020.9468084

8. Iosifova, O., Iosifov, I., Sokolov, V., Romanovskyi, O., & Sukaylo, I. (2021). Analysis of Automatic Speech Recognition Methods. In *Proceedings of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems* (Vol. 2923, pp. 252–257).

9. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Prototyping Methodology of End-to-End Speech Analytics Software. In *Proceedings of the 4th International Workshop on Modern Machine Learning Technologies and Data Science* (Vol. 3312, pp. 76–86).

10. Mahdavifar, S., & Ghorbani, A. (2019). Application of Deep Learning to Cybersecurity: A Survey. *Neurocomputing*, 347, 149–176. https://doi.org/10.1016/j.neucom.2019.02.056

11. Sedkowski, W., & Bierczyński, K. (2022). Perceived Severity of Vulnerability in Cybersecurity: Cross Linguistic Variegation. In *2022 IEEE International Carnahan Conference on Security Technology* (pp. 1–4). https://doi.org/10.1109/iccst52959.2022.9896488

12. Mounnan, O., Manad, O., Boubchir, L., Mouatasim, A., & Daachi, B. (2022). Deep Learning-Based Speech Recognition System using Blockchain for Biometric Access Control. In *2022 9th International Conference on Software Defined Systems (SDS)* (pp. 1–2). https://doi.org/10.1109/SDS57574.2022.10062921

13. Chen, Y., Zhang, J., Yuan, X., Zhang, S., Chen, K., Wang, X., & Guo, S. (2022). SoK: A Modularized Approach to Study the Security of Automatic Speech Recognition Systems. In *ACM Transactions on Privacy and Security* (Vol. 25, no. 3, pp. 1–31). Association for Computing Machinery (ACM). https://doi.org/10.1145/3510582

14. Zhang, S., Gao, Z., Luo, H., Lei, M., Gao, J., Yan, Z., & Xie, L. (2020). Streaming Chunk-Aware Multihead Attention for Online End-to-End Speech Recognition. *Interspeech*. https://doi.org/10.21437/interspeech.2020-1972

15. Kim, S., Seltzer, M., Li, J., & Zhao, R. (2018). Improved Training for Online End-to-End Speech Recognition Systems. *Interspeech* (pp. 2913–2917). https://doi.org/10.21437/interspeech.2018-2517.

16. Chen, Y., Wang, W., Chen, I-F., & Wang, C. (2020). Data Techniques for Online End-to-End Speech Recognition, *arXiv* (pp. 1–5). https://doi.org/10.48550/arXiv.2001.09221

17. Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *8th Workshop on Syntax, Semantics and Structure in Statistical Translation* (pp. 103–111). https://doi.org/10.3115/v1/w14-4012

18. Firth, J. R. (1957). A Synopsis of Linguistic Theory, 1930–1955.

19. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532–1543). https://doi.org/1010.3115/v1/d14-1162

20. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations* (pp. 1–13). https://doi.org/10.48550/arXiv.1301.3781

21. Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologi*es (pp. 2227–2237). https://doi.org/10.18653/v1/n18-1202

22. Iosifova, O., Iosifov, I., Rolik, O., & Sokolov, V. (2020). Techniques Comparison for Natural Language Processing. In *Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS)*, (Vol. 2631, no. I, pp. 57–67).

23. Iosifova, O., Iosifov, I., & Rolik, O. (2020). Methods and Components of Natural Language Processing. In *Adaptive Automatic Control Systems* (Vol. 1, no. 36,

pp. 93–113). Kyiv Politechnic Institute, https://doi.org/10.20535/1560-8956.36.2020.209780

24. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2002). Learning Representations by Back-Propagating Errors. In *Cognitive Modeling* (pp. 213–222). The MIT Press. https://doi.org/10.7551/mitpress/1888.003.0013

25. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS Workshop on Deep Learning and Representation Learning*, *arXiv* (pp. 1–9). https://doi.org/10.48550/arXiv.1412.3555

26. Liu, Q., Kusner, M. J., Blunsom, P. (2020). A Survey on Contextual Embeddings, *arXiv* (pp. 1–13). https://doi.org/10.48550/arXiv.2003.07278

27. Kristoffersen, M. S., Wieland, J. L., Shepstone, S. E., Tan, Z.-H., & Vinayagamoorthy, V. (2019). Deep Joint Embeddings of Context and Content for Recommendation. In *Context-Aware Recommender Systems Workshop* (pp. 1–5). https://doi.org/10.48550/arXiv.1909.06076

28. Zhang, Y., & Ma, Q. (2020). DocCit2Vec: Citation Recommendation via Embedding of Content and Structural Contexts. *IEEE Access* (Vol. 8, pp. 115865–115875). https://doi.org/10.1109/access.2020.3004599

29. Lebret, R., & Collobert, R. (2014). Word Embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Gothenburg, Sweden (pp. 482–490). https://doi.org/10.3115/v1/E14-1051

30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* (Vol. 2, pp. 3111–3119).

31. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press, 462–480.

32. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, MIT Press (Vol. 2, pp. 3104–3112).

33. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Doha, Qatar (pp. 1724–1734). https://doi.org/10.3115/v1/D14-1179

34. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations* (pp. 1–15). https://doi.org/10.48550/arXiv.1409.0473

35. Luong, T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics* (pp. 1412–1421).

36. Kolen, J. F., & Kremer, S. C. (2009). Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies. In *A Field Guide to Dynamical Recurrent Networks*, IEEE (pp. 237–243). https://doi.org/10.1109/9780470544037.ch14

37. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation* (Vol. 9, no. 8, pp. 1735–1780). https://doi.org/10.1162/neco.1997.9.8.1735

38. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine* (Vol. 29, no. 6, pp. 82–97). https://doi.org/10.1109/MSP.2012.2205597

39. Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of Speech Recognition. PTR Prentice Hall, 342–368.

40. Young, S. J., Woodland, P. C., & Byrne, W. J. (1993). HTK – Hidden Markov Model Toolkit, Version 1.5. Cambridge University Engineering Department and Entropic Research Laboratories Inc. (pp. 10–34).

41. McDermott, E. (2018). A Deep Generative Acoustic Model for Compositional Automatic Speech Recognition. In: *32nd Conference on Neural Information Processing Systems* (pp. 1–17).

42. Baum, L. E., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics* (Vol. 37, no. 6, pp. 1554–1563). https://doi.org/10.1214/aoms/1177699147

43. Rabiner, L. R. (1990). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* (Vol. 77, no. 2, pp. 257–286). https://doi.org/10.1109/5.18626

44. Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing* (Vol. 20, no. 1, pp. 30–42). https://doi.org/10.1109/TASL.2011.2134090

45. Jelinek, F. (1976). Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE* (Vol. 64, no. 4, pp. 532–556). https://doi.org/10.1109/proc.1976.10159

46. Poritz, A. (1982). Linear Predictive Hidden Markov Models and the Speech Signal. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing; Institute of Electrical and Electronics Engineers* (Vol. 7, pp. 1291–1294). https://doi.org/10.1109/ICASSP.1982.1171633

47. Wang, D., Wang, X., & Lv, S. (2019). An Overview of End-to-End Automatic Speech Recognition. *Symmetry* (Vol. 11, no. 8, pp. 1–26). https://doi.org/10.3390/sym11081018

48. Zhang, Y., Chan, W., & Jaitly, N. (2017). Very Deep Convolutional Networks for End-to-End Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4845–4849). https://doi.org/10.1109/icassp.2017.7953077

49. Berard, A., Pietquin, O., Servan, C., & Besacier, L. (2016). Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. In *NIPS*

Workshop on End-to-End Learning for Speech and Audio Processing, *arXiv* (pp. 1–5). https://doi.org/10.48550/arXiv.1612.01744

50. Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., & Saurous, R. A. (2017). Tacotron: Towards End-to-End Speech Synthesis. In *Interspeech*. https://doi.org/10.21437/interspeech.2017-1452

51. Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R. A., Agiomvrgiannakis, Y., & Wu, Y. (2018). Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4779–4783). https://doi.org/10.1109/icassp.2018.8461368

52. Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K., & Toda, T. (2017). Speaker-Dependent WaveNet Vocoder. *Interspeech* (pp. 1118–1122). https://doi.org/10.21437/interspeech.2017-314

53. Arik, S., Diamos, G., Gibiansky, A., Miller, J., Peng, K., Ping, W., Raiman, J., & Zhou, Y. (2017). Deep Voice 2: Multi-Speaker Neural Text-to-Speech. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 2966–2974).

54. Baum, L. E., & Eagon, J. A. (1967) An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology, *Bulletin of the American Mathematical Society* (Vol. 73, no. 3, pp. 360–364). https://doi.org/10.1090/s0002-9904-1967-11751-8

55. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Kipchuk, F., & Sukaylo, I. (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 83, pp. 25–36). https://doi.org/10.1007/978-3-030-80472-5_3

56. Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., Schlüter, R., & Ney, H. (2019). RWTH ASR Systems for LibriSpeech: Hybrid vs Attention. *Interspeech* (pp. 1–5). https://doi.org/10.21437/interspeech.2019-1780

57. Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist Temporal Classification. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 369–376). https://doi.org/10.1145/1143844.1143891

58. Hsiao, R., Can, D., Ng, T., Travadi, R., & Ghoshal, A. (2020). Online Automatic Speech Recognition with Listen, Attend and Spell Model. *IEEE Signal Processing Letters* (Vol. 27, pp. 1889–1893). https://doi.org/10.1109/lsp.2020.3031480

59. Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1–16). https://doi.org/10.1109/icassp.2016.7472621

60. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North. Association for Computational Linguistics* (pp. 1–16). https://doi.org/10.18653/v1/n19-1423

61. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach, *arXiv* (pp. 1–13). https://doi.org/10.48550/arXiv.1907.11692

62. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations, *arXiv* (pp. 1–17). https://doi.org/10.48550/arXiv.1909.11942

63. Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-Training Text Encoders as Discriminators Rather than Generators, *arXiv* (pp. 1–18). https://doi.org/10.48550/arXiv.2003.10555

64. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, *arXiv* (pp. 1–5). https://doi.org/10.48550/arXiv.1910.01108

65. Lample, G., & Conneau, A. (2019). Cross-Lingual Language Model Pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Article 634, pp. 7059–7069).

66. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models Are Unsupervised Multitask Learners. *OpenAI* (Vol. 1, pp. 1–24).

67. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2978–2988). https://doi.org/10.18653/v1/p19-1285

68. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q.V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Article 517, pp. 5753–5763).

69. Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., & Catanzaro, B. (2020). Megatron-LM: Training Multi-Billion Parameter Language Models using Model Parallelism, *arXiv* (pp. 1–15). https://doi.org/10.48550/arXiv.1909.08053

70. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. K., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., & Veselý, K. (2011). The Kaldi Speech Recognition Toolkit. In *Proceedings of the ASRU* (pp. 1–4).

71. Lee, A., Kawahara, T., & Shikano, K. (2001). Julius—an Open Source Real-Time Large Vocabulary Recognition Engine. In *7th European Conference on Speech Communication and Technology (Eurospeech)* (Vol. 1–4). https://doi.org/10.21437/eurospeech.2001-396

72. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A.Y. (2014). Deep Speech: Scaling up End-to-End Speech Recognition, *arXiv* (pp. 1–12). https://doi.org/10.48550/arXiv.1412.5567

73. Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). ESPnet: End-to-End Speech Processing Toolkit. In *Interspeech* (pp. 2207–2211). https://doi.org/10.48550/arXiv.1804.00015

74. Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). Wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Interspeech* (pp. 3465–3469). https://doi.org/10.21437/interspeech.2019-1873

75. Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020) Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Article 1044, pp. 12449–12460).

76. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 1–8). https://doi.org/10.18653/v1/2020.emnlp-demos.6

77. Tan, H., Wang, L., Zhang, H., Zhang, J., Shafiq, M., & Gu, Z. (2022). Adversarial Attack and Defense Strategies of Speaker Recognition Systems: A Survey. In *Electronics* (Vol. 11, no. 14, p. 2183). MDPI AG. https://doi.org/10.3390/electronics11142183

РОЗДІЛ 2
ПІДХОДИ ДО ПІДВИЩЕННЯ БЕЗПЕКИ ТА ЕФЕКТИВНОСТІ РОЗПІЗНАВАННЯ ГОЛОСОВОЇ ІНФОРМАЦІЇ

2.1. Підходи до забезпечення безпеки голосової інформації

2.1.1. Структура інформаційних систем та кіберзагрози

Сучасні загрози характеризуються своєю складністю і інтегрованістю в декілька каналів одночасно. Атаки з використанням голосової інформації можуть бути виконані як ізольовані, наприклад з використанням рацій або телефонів, так і інтегровані в інші канали, такі як соціальні мережі або з частковим використанням електронної пошти як додаткового каналу передачі інформації [1].

На рис. 2.1 зображена спрощена структура виникнення і захисту від кіберзагроз.

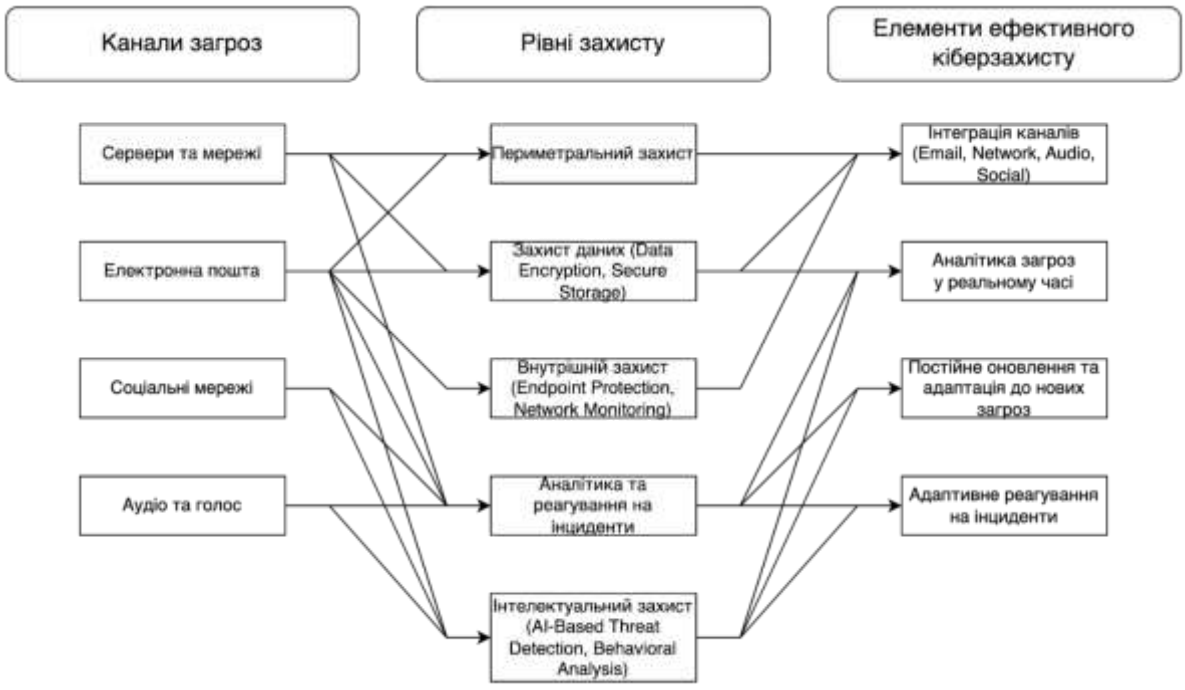


Рис. 2.1. Взаємозв’язок каналів загроз, рівнів захисту та елементів кіберзахисту

Це охоплює кілька ключових компонентів для забезпечення безпеки організацій і державних установ, а саме *канали загроз*, через які можуть виникати загрози і *рівні захисту*, що можуть бути застосовані для кожного з каналів загроз. Оскільки елементи рівнів захисту, зазвичай, не пов'язані між собою – це ускладнює виявлення і захист від мультиканальних загроз. Для виявлення сучасних, мультиканальних загроз потрібен інтегрований рівень захисту, що відображається у блоці елементи ефективного кіберзахисту і демонструє, як різні рівні захисту можуть бути інтегровані в єдину систему для комплексного забезпечення безпеки.

2.1.2. Місце та роль голосової інформації при забезпеченні захисту від кіберзагроз

Одним з найскладніших для аналізу і водночас найбільш натуральним є канал голосової інформації. Аудіоінформація вбудована в усі сфери життя людини і тим чи іншим чином в більшість каналів придатних для атаки. Розглянемо детальніше рівні загроз і можливі системи захисту від них. На рис. 2.2 представлена деталізація систем і підходів для різних рівнів відображає (зелений колір) системи, пов'язані з захистом або ризиком атак використовуючи аудіо канал.

Рис. 2.2. Елементи кібербезпеки в системах розпізнавання голосової інформації

На державному рівні для боротьби з загрозами на основі аудіо даних використовується цілі спектри систем, платформ і підходів, які відіграють важливу роль у забезпеченні кібербезпеки та захисту від загроз у сучасному інформаційному просторі. Кожен з елементів цієї схеми виконує свою унікальну функцію, яка доповнює інші, створюючи таким чином комплексну структуру захисту.

Розглянемо системи масового прослуховування та аналізу, які є критичними для виявлення потенційних загроз на ранніх етапах. Вони дозволяють державним та правоохоронним органам отримувати доступ до великого обсягу даних з різних джерел комунікації та аналізувати ці дані з метою виявлення підозрілих шаблонів поведінки або загроз. Ці системи часто використовуються для боротьби з тероризмом та іншими видами злочинної діяльності. Але як видно з назви (масового) ці системи доволі обширні, що відображається у вартості їх роботи (обчислювальні ресурси, електрика, тощо). Тому в паралель використовуються Системи цільового моніторингу орієнтовані на спостереження за конкретними об’єктами або групами людей. Вони використовуються для стеження за підозрюваними особами, забезпечення безпеки важливих подій або об’єктів, а також для запобігання можливим загрозам у реальному часі. Ці системи дозволяють зосередитися на конкретних цілях і забезпечують детальний аналіз їх діяльності. Для забезпечення безпеки важливих подій, об’єктів або систем додатково використовуються системи ідентифікації за біометричними даним, такі як відбитки пальців, розпізнавання обличчя або голосу, для підтвердження особи користувачів. Вони забезпечують високий рівень безпеки у порівнянні з традиційними методами аутентифікації, такими як паролі. Ці системи широко застосовуються у правоохоронних органах, банківському секторі та для контролю доступу до важливих об’єктів.

Оскільки масовий моніторинг і прослуховування не є достатнім в поточному світі де соціальні мережі відіграють значну роль у поширенні інформації та формуванні суспільної думки, важливим елементом також є платформи аналізу соціальних мереж, що спеціалізуються на моніторингу та аналізі активності в

соціальних мережах [2, 3]. З їх допомогою можна виявляти дезінформаційні кампанії, маніпуляції громадською думкою, а також слідкувати за підозрілими акаунтами або групами. На більш низькому рівні роботи з системами масового і цільового прослуховування а також з платформи аналізу соціальних мереж використовуються інструменти лінгвістичного аналізу. Вони допомагають аналізувати текстові дані для виявлення ключових слів, настроїв або прихованих повідомлень. Вони використовуються як у кібербезпеці, так і в контексті розвідки та контррозвідки. Ці інструменти дозволяють отримувати цінну інформацію з текстових джерел, таких як соціальні мережі, електронна пошта або інші види комунікацій.

Системи моніторингу збирають і структурують інформацію з різних каналів і джерел для виявлення атак і зловмисників по зібраним даним використовуються Системи виявлення аномалій, які є важливою частиною будь-якої стратегії кібербезпеки. Вони використовують методи ML та аналізу даних для виявлення незвичайної активності у мережевому трафіку або поведінці користувачів. Виявлення аномалій може бути першим кроком у виявленні кібератак або інших загроз, які не були зафіксовані традиційними засобами безпеки.

Як було зазначено вище, сучасні методи і вектори атак використовують одночасно багато каналів і джерел, тож дані і виявлені аномалії повинні бути інтегровані в більш високорівневі інтегровані платформи кібербезпеки, що об’єднують кілька різних компонентів захисту, таких як захист від кіберзагроз, аналітика загроз у реальному часі та можливості реагування на інциденти. Ці платформи здатні забезпечити цілісну картину безпеки для організацій, дозволяючи їм контролювати всі аспекти кібербезпеки з одного місця. Вони важливі для великих організацій, які потребують комплексного підходу до управління безпекою та ефективного реагування на різноманітні загрози.

Таким чином, всі ці системи і платформи взаємодіють між собою, створюючи комплексну структуру, яка забезпечує ефективний захист як від кіберзагроз, так і від фізичних загроз з боку зловмисників. Вони відіграють ключову роль у сучасних

стратегіях безпеки, як на рівні організацій, так і на державному рівні. Так можна виділити кілька різних типів систем, наприклад:

1. Системи масового прослуховування та аналізу:

– ECHELON – глобальна система радіоелектронної розвідки;

– PRISM – програма збору та аналізу даних електронних комунікацій;

– XKeyscore – система пошуку та аналізу глобальних інтернет-даних.

2. Системи цільового моніторингу:

– Carnivore/DCS1000 – система моніторингу електронної пошти;

– Stingray– пристрої для перехоплення мобільних комунікацій.

3. Платформи аналізу соціальних мереж:

– Palantir – платформа для аналізу великих даних та виявлення зв’язків;

– Babel Street – система моніторингу та аналізу соціальних медіа.

4. Системи біометричної ідентифікації:

– IDENT– система біометричної ідентифікації США;

– VoiceGrid – система розпізнавання голосу для правоохоронних органів.

5. Інструменти лінгвістичного аналізу:

– VADER (Valence Aware Dictionary and sEntiment Reasoner) – інструмент для аналізу настроїв;

– LIWC (Linguistic Inquiry and Word Count) – програма для аналізу тексту.

6. Системи виявлення аномалій:

– NIST (National Institute of Standards and Technology) Anomaly Detection – система для виявлення аномалій у великих наборах даних;

– IBM QRadar – платформа для виявлення загроз та аномалій у мережевому трафіку.

7. Інтегровані платформи кібербезпеки:

– IBM i2 Analyst’s Notebook – платформа для аналізу та візуалізації даних;

– Splunk – платформа для аналізу машинних даних та виявлення загроз.

Варто зазначити, що детальна інформація про багато систем, що використовуються державними органами, часто є засекреченою, але технології постійно розвиваються і нові системи та підходи з’являються регулярно.

2.1.3. Перспективи застосування обробки природної мови в кібербезпеці

У сучасному цифровому просторі, де обсяг інформації та складність кіберзагроз постійно зростають, кібербезпека стає одним із найважливіших напрямів досліджень та практичного застосування. Сучасні технології NLP та GPT-архітектури відкривають нові можливості для виявлення, аналізу та запобігання кіберзагрозам, що робить їх актуальними для використання в цій сфері.

Інтеграція сучасних NLP та GPT-архітектур у сферу кібербезпеки є новим та актуальним напрямом, що пропонує значні переваги у боротьбі з кіберзагрозами. Ці технології дозволяють глибше розуміти та аналізувати текстову інформацію, виявляючи складні шаблони та аномалії. Однак для максимально ефективного їх використання необхідно вирішувати існуючі виклики та забезпечувати етичне та правове обґрунтування їх застосування.

Традиційні методи кібербезпеки часто базуються на сигнатурному аналізі та наборах правил, які не завжди здатні ефективно виявляти нові або модифіковані загрози. Сучасні NLP-моделі, зокрема ті, що базуються на GPT-архітектурах, здатні аналізувати великі обсяги текстових даних та виявляти складні шаблони, які можуть вказувати на потенційні загрози.

Новизна цих підходів полягає у використанні глибокого навчання та трансформерних моделей для обробки природної мови, що дозволяє системам розуміти контекст та семантику тексту на більш високому рівні. Це особливо корисно для:

– виявлення фішингових атак (GPT-моделі можуть аналізувати електронні листи та повідомлення, виявляючи нетипові мовні конструкції та зміст, що характерні для фішингу);

– аналізу шкідливого програмного забезпечення (розуміння коду та коментарів у шкідливих програмах допомагає у їх швидкому виявленні та нейтралізації);

– моніторингу темних веб-ресурсів (NLP може використовуватися для аналізу дискусій та повідомлень на форумах, де кіберзлочинці обмінюються інформацією).

Зі зростанням кількості кібератак та їх складності, традиційні методи захисту стають менш ефективними. Сучасні NLP та GPT-моделі здатні адаптуватися до нових загроз завдяки своїй архітектурі та можливості навчатися на великих наборах даних. Це робить їх незамінними у таких аспектах:

– можуть швидко аналізувати нову інформацію та оновлювати свої алгоритми виявлення;

– можуть бути налаштовані під специфічні потреби організації, враховуючи унікальні шаблони поведінки та потенційні ризики;

– дозволяють автоматизувати багато рутинних задач, зменшуючи навантаження на аналітиків та підвищуючи ефективність роботи.

Незважаючи на значні переваги, використання сучасних NLP та GPT-архітектур у кібербезпеці має ряд викликів:

– для ефективного навчання моделей необхідні великі набори даних, що можуть бути недоступними або конфіденційними;

– ті ж самі моделі можуть бути використані кіберзлочинцями для створення більш переконливих фішингових атак або шкідливого контенту;

– обробка персональних даних вимагає дотримання законодавства про конфіденційність та захист інформації.

Сьогодні дослідження у сфері NLP зосереджені на покращенні моделей трансформерів, розробці ефективніших алгоритмів та розширенні застосувань на різні мови та домени. З’являються моделі з мільярдами параметрів, такі як Claude Sonnet та GPT-4, які демонструють вражаючу здатність до генерації тексту, розуміння контексту та виконання складних завдань без спеціального навчання.

NLP відіграє кілька важливих ролей у сучасній кібербезпеці. Вдосконалені алгоритми NLP можуть аналізувати шаблони в текстових повідомленнях, щоб виявити потенційні спроби фішингу, визначаючи підозрілі мовні шаблони, незвичні запити або тактики соціальної інженерії. Команди безпеки використовують NLP для автоматичної обробки та категоризації журналів безпеки та сповіщень, що допомагає їм визначати пріоритети та ефективніше реагувати на загрози.

Також NLP дає змогу аналізувати настрої комунікацій у темному інтернеті, щоб виявляти нові загрози та відстежувати діяльність кіберзлочинців. Завдяки класифікації тексту та виявленню аномалій NLP допомагає ідентифікувати підозрілі послідовності команд та потенційні спроби виконання шкідливого коду. Системи на основі NLP можуть відстежувати внутрішні комунікації на предмет потенційного витоку даних або внутрішніх загроз, відзначаючи незвичні шаблони спілкування або несанкціонований обмін конфіденційною інформацією.

Чат-боти з NLP можуть забезпечити негайне реагування першого рівня на інциденти безпеки та провести користувачів через протоколи безпеки. Методи NLP допомагають аналізувати описи шкідливих програм і звіти про загрози, щоб виявити схожість між новими і відомими загрозами, допомагаючи в розвідці загроз. Обробляючи запити природною мовою, NLP робить документацію з безпеки та бази знань більш доступними для команд безпеки під час реагування на інциденти.

Аналіз мережевого трафіку виграє від NLP завдяки виявленню командно-контрольних комунікацій і спроб витоку даних, прихованих у трафіку, що виглядає легітимно. Системи захисту електронної пошти використовують NLP для виявлення спаму, спроб компрометації ділової електронної пошти та складних фішингових кампаній. Вдосконалені моделі NLP можуть аналізувати коментарі до коду та документацію, щоб виявити потенційні вразливості безпеки під час процесу розробки.

Дотримання політики безпеки можна автоматизувати, використовуючи NLP для сканування документів і повідомлень на предмет потенційних порушень. NLP допомагає обробляти та співвідносити дані про загрози з різних джерел для створення комплексних оцінок загроз. Можливості генерації природної мови допомагають створювати автоматизовані звіти та оповіщення про безпеку, які є більш зрозумілими та зручними для команд безпеки. Аналізуючи взаємодію з людиною, NLP може виявити спроби несанкціонованого доступу або незвичайні моделі поведінки, які можуть свідчити про компрометацію.

Тренінги з підвищення обізнаності про безпеку виграють від NLP завдяки персоналізованій подачі контенту та оцінці розуміння користувачами. Системи на

основі NLP можуть автоматично генерувати і підтримувати документацію з безпеки, обробляючи звіти про інциденти і процедури реагування. Полювання на загрози стає більш ефективним, оскільки NLP допомагає аналітикам швидко обробляти і співвідносити величезні обсяги неструктурованих даних про безпеку. Системи розпізнавання голосу, вдосконалені за допомогою NLP, можуть додати додатковий рівень безпеки для голосових систем і виявляти потенційні голосові атаки.

Проте існують серйозні виклики, пов’язані з ресурсомісткістю моделей, багатомовністю, інтерпретованістю та етичними аспектами, тому для подальшого розвитку та впровадження NLP технологій необхідно:

– розроблювати ефективніші алгоритми та архітектури, що зменшують потребу в обчислювальних ресурсах (наприклад, використання знань дистиляції, прюнінгу та квантізації);

– розширювати дослідження на інші мови, особливо ті, що мають обмежені ресурси, та використовувати методи переносу знань та багатомовних моделей;

– покращувати розуміння внутрішніх механізмів моделей для підвищення довіри та можливості пояснити результати та розроблювати інструменти для візуалізації та аналізу рішень моделей;

– врахувати приватності, упередженості та інших етичних питань при розробці та впровадженні NLP технологій і встановлювати стандарти та практики для відповідального використання штучного інтелекту;

– розроблювати методи, що дозволяють використовувати NLP технології на пристроях з обмеженими ресурсами, таких як мобільні телефони або пристрої інтернету речей.

Але актуальність досліджень у сфері NLP підкріплюється економічними вигодами. Інвестиції в технології NLP можуть привести до значного економічного зростання, створення нових робочих місць та підвищення продуктивності в різних секторах економіки. Компанії, що впроваджують передові NLP-рішення, отримують конкурентні переваги, можуть пропонувати інноваційні продукти та послуги, задовольняючи зростаючі потреби ринку [4].

2.1.4. Підходи до побудови системи інформаційної безпеки, яка працює з голосовою інформацією

Є два підходи до побудови системи безпеки сфокусованій на голосовій інформації:

1. Агрегування максимально можливої інформації з максимальної можливої кількості існуючих систем та продуктів (див. табл. 2.1), передбачає встановлення і інтеграцію даних із різноманітних джерел, таких як аудіосистеми, системи відеоспостереження, сенсори руху, системи ідентифікації та інші засоби моніторингу. Метою є створення єдиної платформи, яка може зібрати якомога більше інформації для всебічного аналізу, виявлення аномалій, загроз та прийняття рішень у системі безпеки.

Таблиця 2.1

Порівняльний аналіз підходу агрегації інформації

Переваги	Недоліки
1. *Широкий спектр даних*, тому отримується найбільш повна картина того, що відбувається, оскільки система аналізує дані з різних джерел. 2. *Висока точність*, бо інтеграція даних з різних систем зменшує ймовірність помилкових тривог, оскільки система має більше інформації для аналізу. 3. *Універсальність* дозволяє використовувати даний підхід до різних типів загроз та середовищ.	1. *Вартість*, бо агрегування даних з різних систем вимагає значних фінансових та технічних ресурсів, але підтримка в актуальному стані ще більш дорогий процес аніж первинна інтеграція. 2. *Інтеграція* великої кількості систем може бути технічно складною і вимагати багато часу для налаштування та підтримки. 3. *Обмеження релевантних даних* за рахунок значних обмежень у кастомізації того, які саме дані і в якому форматі збираються, з якими параметрами і якими моделями. 4. *Конфіденційність*, бо деякі системи є хмарними та пропрієтарними із закритим вихідним кодом і архітектурою.

2. Створення системи під кожну конкретну проблему передбачає розробку вузькоспеціалізованих систем безпеки, що зосереджуються на вирішенні конкретних завдань або проблем. Наприклад, для захисту від голосових загроз у колл-центрі може бути розроблена система, яка фокусується лише на розпізнаванні аномалій у голосах клієнтів або голосової ідентифікації/аутентифікації.

Цей підхід також несе в собі характеристики так званої «клаптикової» інтеграції (див. табл. 2.2), але в даному підході нівелюються недоліки першого підходу пов’язані з гнучкістю, швидкістю і адаптивністю системи.

Таблиця 2.2

Порівняльний аналіз підходу індивідуальних систем

Переваги	Недоліки
1. *Ефективність*, оскільки системи спеціально налаштовані на виявлення конкретних загроз. 2. *Нижча вартість* системи за рахунок менших затрат на розробку та впровадження, оскільки вони мають менший обсяг даних для обробки. 3. *Простота реалізації*, бо вузькоспеціалізовані системи простіше налаштовувати та підтримувати.	1. *Обмежена функціональність*, бо така система може бути ефективною лише в певних сценаріях і не зможе забезпечити загальний захист. 2. *Низька адаптивність* до нових загроз може потребувати додаткових налаштувань або навіть повної модернізації. 3. *Фрагментація даних*, бо використання різних систем для різних завдань може призвести до проблем з координацією та управлінням всією системою безпеки.

Системи аналізу намірів і загроз у голосовій інформації повинні бути реалізовані у вигляді *рекомендаційної* або *консультативної* системи. Вирішуючі та дорадчі системи не підходять для вирішення задачі, оскільки вони можуть допускати помилки, які можуть призвести до серйозних наслідків.

Рекомендаційні системи надають користувачам можливість самостійно приймати остаточні рішення на основі аналізу даних, що знижує ризик помилкових дій з боку автоматизованої системи. Консультативні системи, в свою чергу,

дозволяють отримувати поради та пропозиції щодо можливих дій, залишаючи за користувачем максимальний контроль над кінцевим вибором.

Вибір на користь таких підходів пов’язаний з необхідністю мінімізувати можливість хибних спрацьовувань, які можуть виникнути через складність інтерпретації голосових даних. Системи, що приймають остаточні рішення, можуть помилятися в критичних ситуаціях, що робить їх менш надійними у порівнянні з системами, які лише надають рекомендації або поради.

Таким чином, системи аналізу голосової інформації повинні фокусуватися на підтримці користувача в прийнятті рішень, генеруючи нотифікації, надаючи їм необхідні дані та рекомендації, але залишаючи останнє слово за людиною. Це забезпечить більшу гнучкість і знизить ризик негативних наслідків від помилкових дій системи.

Реалізація рекомендаційної системи виявлення зловмисних намірів і атак в голосових даних має включати:

– збір даних з різних джерел;

– попередню обробку та фільтрацію даних;

– застосування алгоритмів ML та NLP;

– аналіз результатів експертами;

– інтеграцію з іншими системами безпеки.

Для побудови таких систем існує ряд платформ, моделей, прототипів та програм, що працюють з голосовими даними. Основними системами та платформами для розпізнавання голосу є:

– Nuance Communications Dragon – система розпізнавання голосу, яка використовується у багатьох галузях, включаючи медицину та юриспруденцію [5];

– Google Speech-to-Text – потужна хмарна платформа для перетворення голосу в текст, що підтримує численні мови та акценти [6];

– Microsoft Azure Speech Services – хмарний сервіс, який пропонує функції розпізнавання та синтезу мовлення з можливостями персоналізації моделей для конкретних сценаріїв використання [7];

– Amazon Transcribe – сервіс від Amazon Web Services для автоматичного транскрибування голосових файлів у текст з можливістю інтеграції в різні бізнес-процеси [8];

– IBM Watson Speech-to-Text – рішення від IBM для розпізнавання мовлення, яке підтримує кілька мов і спеціалізується на інтеграції з іншими ШІ-сервісами Watson [9].

Також слід зазначити існуючі прототипи та інструменти для досліджень:

– Kaldi – відкрита платформа для створення систем розпізнавання мовлення, що активно використовується в наукових дослідженнях та експериментальних проєктах [10];

– DeepSpeech (Mozilla) – відкрита система розпізнавання мовлення на основі DL, яка має за мету забезпечити доступність технологій розпізнавання голосу для всіх [11];

– Julius – система розпізнавання мовлення з відкритим вихідним кодом, яка використовується у різних проєктах з обробки мовлення [12];

– Pocketsphinx – легка система розпізнавання мовлення, що підходить для інтеграції у мобільні та вбудовані системи [13].

Окремим підвидом є біометричні системи ідентифікації за голосом:

– BioID Voice Recognition – системи біометричної ідентифікації, які використовують голосовий відбиток для підтвердження особи [14];

– Agnitio Voice ID – рішення для біометричної ідентифікації за голосом, яке використовується у безпеці та правоохоронних органах [15].

Крім того, для аналізу емоцій та інтонацій існують окремі інструменти:

– Beyond Verbal – платформа, що аналізує емоційний стан людини на основі її голосу [16];

– Affectiva – система аналізу емоцій, яка може використовувати голос для оцінки емоційного стану у реальному часі [17].

А також існують інші інструменти для обробки голосових даних, наприклад:

– VoxSigma – система для автоматичного транскрибування та аналізу аудіо-файлів, що використовується для моніторингу ЗМІ та юридичних досліджень [18];

– Speechmatics – сервіс для розпізнавання голосу, який підтримує транскрибування на кількох мовах та використовується у медіа, телекомунікаціях та інших галузях [19].

Цей перелік включає як комерційні, так і відкриті рішення, що дозволяє вибрати відповідну технологію залежно від конкретних потреб та ресурсів організації.

На рис. 2.3 зображено схему процесу аналізу текстових та голосових даних для виявлення злочинних та небезпечних намірів.

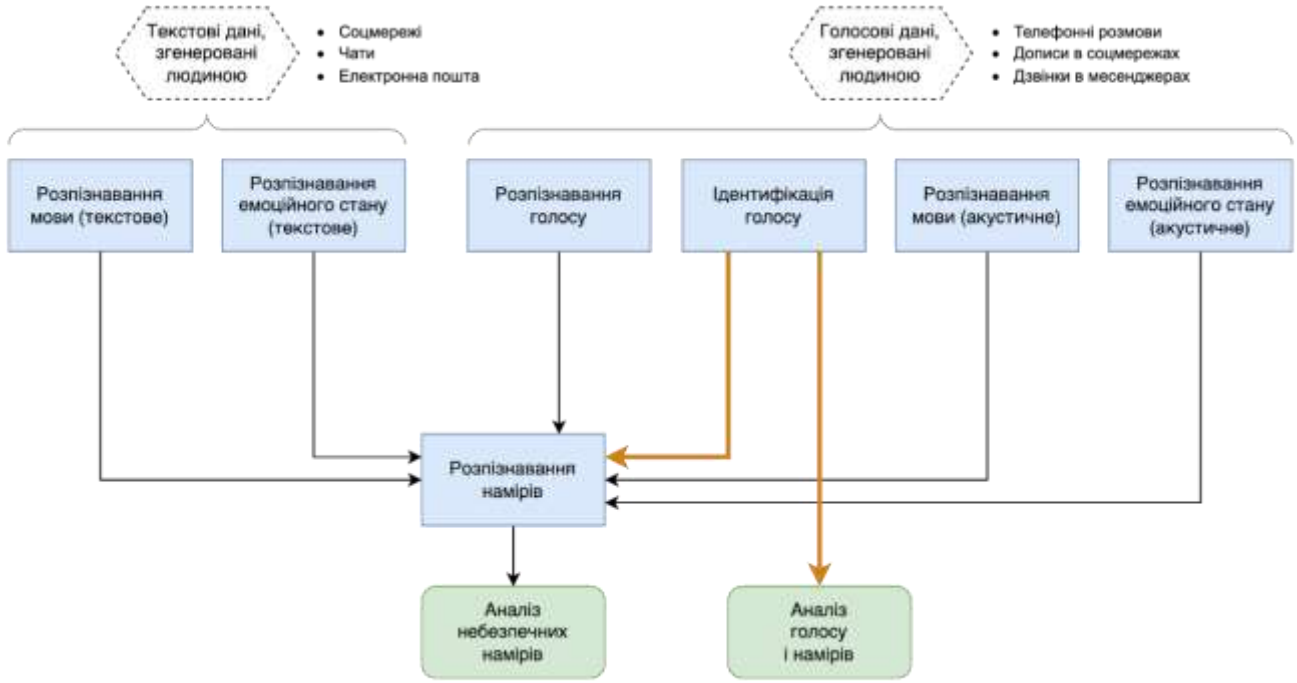


Рис. 2.3. Порядок взаємодії систем
розпізнавання голосової і текстової інформації для аналізу намірів

Текстові та голосові дані, згенеровані людиною, проходять через різні етапи розпізнавання: мови, емоційного стану (як текстового, так і акустичного), голосу та ідентифікації голосу [20]. На основі цих даних здійснюється розпізнавання намірів, що дозволяє проводити аналіз небезпечних намірів або голосу з додатковою оцінкою можливих загроз [21].

2.2. Метрики оцінювання та критерії вимірювання якості розпізнавання

2.2.1. Метрики обробки природної мови

Оскільки сфера NLP досить широка, а кількість завдань в NLP дуже велика – не існує єдиної загальної метрики для всіх завдань. Ми можемо розділити метрики, кластеризувавши завдання і виділивши наступні:

– моделі машинного перекладу: двомовний оцінювальний дублер – це показник ефективності для вимірювання продуктивності моделей машинного перекладу. Він оцінює, наскільки добре модель перекладає з однієї мови на іншу;

– оцінювання розуміння мови: загальне оцінювання розуміння мови є критерієм, що ґрунтується на різних типах завдань, а не на оцінюванні одного завдання. Три основні категорії завдань – це завдання на розуміння одного речення, завдання на встановлення подібності та перефразування, а також завдання на умовивід.

Водночас є багато завдань з точного налаштування, як-от тегування частини речення, розпізнавання іменованих сутностей тощо. У таких завданнях найпоширенішою метрикою є підрахунок точності через точність

$$P = \frac{N_{tp}}{N_{tp} + N_{fp}}, \tag{2.1}$$

де N_{tp} – істинних спрацьовувань, N_{fp} – хибних спрацьовувань.

І нагадаємо, що коефіцієнт розраховується за рівністю

$$R = \frac{N_{tp}}{N_{tp} + N_{fn}}, \tag{2.2}$$

де N_{fn} – хибнонегативний результат.

Оцінка F1 розраховується за формулами (2.1) і (2.2) [22]:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}, \tag{2.3}$$

2.2.2. Критерії вимірювання якості обробки природної мови

Придумати єдину метрику для ASR набагато простіше, ніж для завдань NLP, оскільки нам потрібно лише виміряти, чи правильно розпізнано слово, чи ні. Тому WER – найпоширеніша метрика точності ASR. Чим нижчий WER, тим краща система ASR. WER можна обчислити як:

$$WER = \frac{N_S + N_D + N_I}{N_S + N_D + N_C}, \quad (2.4)$$

де N_S – кількість підстановок, N_D – кількість вилучень, N_I – кількість вставок, N_C – кількість правильних слів [23].

Варто зазначити, що WER дуже чутливий до домену та акустики. Наприклад, низький (хороший) WER у 5% для домену літератури (орієнтованого на книги) може мати 20–30% WER для дзвінків у колл-центр.

2.3. Метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів

Мовленнєві асистенти стали більш широко використовуватися людьми [24–26]. Це призводить до збільшення кількості корпорацій, стартапів та дослідницьких груп, готових створювати моделі ASR. Ця готовність стикається з основним обмеженням: брак доступних наборів даних для навчання моделей ASR, що особливо критично для мов з низьким рівнем ресурсів і специфічних областей застосування бажаної моделі ASR (наприклад, медицина, фінанси, страхування і т. ін.). Це обмеження також перешкоджає залученню інженерів-програмістів у сферу розпізнавання мови. Щоб усунути це обмеження, командам доводиться виконувати кілька кроків: отримати набір відповідних аудіозаписів; отримати або створити транскрипцію для всіх або частини аудіозаписів (найскладніша і найдорожча операція); очистити дані; перетворити їх у формат, підтримуваний інструментом для навчання моделі; навчити модель [27].

Великі компанії здебільшого публікують наукові статті та проєкти з відкритим вихідним кодом, які зосереджені на навчанні моделей, що можуть досягти нижчого рівня помилок, використовуючи для навчання та перевірки один з публічних наборів даних [28, 29]. З іншого боку, побудова ефективного конвеєра для підготовки наборів даних є критично важливою частиною отримання високоякісної моделі ASR. Отже, існує потреба в інструменті, який охоплює або етапи попереднього навчання, або всі етапи.

Оскільки якість навченої моделі залежить від кількості аудіоданих, кількість аудіоданих дуже важлива. Але велика кількість даних є дуже дорогою. У [30] проблема недостатньої кількості даних вирішується шляхом збільшення набору даних до 53 000 год., що економить багато ресурсів для роботи. Але цей підхід вимагає попереднього навчання моделей шляхом маскування частини даних з часовими кроками вихідних даних, підготовки конвеєра для нерозмічених даних та певних ресурсів.

Робота [31] присвячена порівнянню підходів до навчання на контрольованих і неконтрольованих даних. Дослідження різних навчальних мовних моделей показує реальну продуктивність методів точного розпізнавання мови. Запропонована стратегія полягає в тому, що наступна година нерозмічених даних навчається на одній годині даних, навчених вручну, а потім ця операція повторюється шляхом подвоєння. Непідконтрольні дані вимагають величезної кількості нерозміченого аудіо та відповідних ресурсів для підготовки та навчання моделей, які відсутні в українській мові, що і стало причиною вибору підконтрольних даних.

Практичне застосування статті [32] є гарним прикладом використання одного з найбільших ресурсів youtube.com для реалізації ШНМ для навчання глибоких нейронних мереж і створення відеосубтитрів з аудіоданих. Правильність вирівнювання тексту до аудіо розраховується за допомогою коефіцієнта «впевненості», який частково покладається на ручне навчання. Розпізнавання мовлення з напівконтрольованою автосинхронізацією даних показує кращі результати і є важливим для правильного розпізнавання та вирівнювання мовлення для неконтрольованих медіа-даних.

2.3.1. Порівняння зі спорідненими інструментами та фреймворками

При запуску нової моделі одним з основних обмежуючих факторів є час. Це може бути як пряме обмеження (наявний час на дослідження), так і непряме (період, який охоплює інвестиції в розробку продукту). Ми визначили два фактори, які можна оптимізувати. Перший фактор – це час для старту. Існуючі інструменти мають круту криву навчання (наприклад, Kaldi) і припускають наявність набору даних у попередньо визначеному форматі (наприклад, wav2letter). З нашого досвіду та обговорень зі спільнотою, нерідко доводиться витрачати дні, а в деяких випадках і тижні перед початком першого навчання на моделі. Якщо перший фактор потрібно виконати один раз, то другий – повторюваний. Створення користувацького набору аудіоданих – це безперервний процес з додаванням нових даних, вилученням неякісних аудіозаписів тощо. У такому сценарії обробка всіх аудіо після кожної зміни уповільнює швидкість експериментів. Якщо ми можемо обробити лише змінену частину – це дозволить провести більше експериментів і зменшить час обчислень та енергоспоживання.

Ми визначили мінімальний набір вимог до конвеєрної системи для усунення факторів уповільнення:

1. Підтримка побудови *конвеєра попередньої обробки*.
2. Підтримка даних з *різних джерел* у *різних форматах*.
3. Підтримка *збереження та анулювання* результатів попередньої обробки.
4. *Першокласна інтеграція* з перетворенням тексту та аудіо.
5. *Низька крива навчання*.

Щоб зробити це формально, ми визначимо набір факторів, які зменшують час навчання:

1. Єдина мова програмування (деякі інструменти використовують суміш різних мов, таких як Bash, Perl, Python, C++, але це збільшує час налаштування середовища).

2. Підтримка специфічних для ASR процедур (для аудіо та тексту) з коробки або наявність задокументованого способу їх інтеграції.

3. Підтримка сучасних інтегрованих середовищ розробки (від англ. Integrated Development Environment, IDE), а також прикладний програмний інтерфейс (від англ. Application Programming Interface, API).

Наведений вище перелік може бути використаний як критерій для оцінки існуючих інструментів, а також як набір вимог до розроблюваної конвеєрної системи.

У табл. 2.3 представлений порівняльний аналіз найпоширеніших існуючих SPT та інструментів для побудови моделі ASR, включаючи NLP інструменти, які не мають прямого відношення до ASR, але пов'язані з попередньою обробкою даних, такі як Text Toolkit Huggingface для нормалізації та Deep Learning Framework Tensorflow/Pytorch для навчання графемно-фонемної моделі.

Як видно з порівняльного аналізу, жоден інструментарій не може впоратися з усіма етапами попередньої обробки даних ASR (включаючи збір, розбиття, маркування, підготовку до очікуваного формату ASR) та навчання моделей.

2.3.2. Модель автоматизованого конвеєра

Як було визначено в попередньому розділі, використання однієї мови програмування є фактором, що зменшує час навчання. Крім того, ми припускаємо, що в команді ШНМ, ймовірно, будуть працювати фахівці з даних та інженери з ML, тому було б корисно використовувати мову програмування, з якою вони добре знайомі. В результаті ми вирішили використовувати Python для фреймворку пайплайну. Ми вирішили зосередитися на малих і середніх наборах даних (до однозначних тисяч годин). Розмір даних, який можна обробити на одній машині, дозволяє нам спростити використання пайплайну. Ми вважаємо це прийнятним компромісом, оскільки нові дослідники та команди навряд чи матимуть більше 10 000 год. аудіо.

Таблиця 2.3

Порівняльний аналіз існуючого програмного забезпечення

Продукт	Тип	Специфічні процедури	Кеш
Kaldi	Інструментарій ASR	Широкий вибір скриптів для обробки аудіо та текстових даних; тренувальні процедури	Повторне використання результатів вручну, ручна ануляція
Julius	Гібридна	Покладається на зовнішні інструменти для попередньої обробки даних	Не підтримується
DeepSpeech	Інструментарій ASR	Функції побудови LM та підпрограми вилучення функцій для аудіо	Підтримується лише для MFCC
EspNet	Гібридна	Широкий вибір скриптів для обробки аудіо та текстових даних; тренувальні процедури	Повторне використання результатів вручну, ручна ануляція
Wav2Vec	Інструментарій ASR	Широкий вибір процедур попередньої обробки аудіо та текстових даних	Надані утиліти для кешування в пам’яті
Hugging Face	Наскрізна	Токенізатори та нормалізація тексту	Підтримується, без часткової недійсності
Pytorch	Інструментарій ASR	Широкий вибір процедур попередньої обробки аудіо та текстових даних у текстових та аудіо модулях	Не підтримується
TensorFlow	Наскрізна	Невеликий набір процедур попередньої обробки тексту, широкий спектр процедур обробки сигналів для аудіо	Підтримується через API знімків, без часткової недійсності

Як показано на рис. 2.4, ми визначили чотири основні типи компонентів:

– DataProxy – об’єкт для передачі даних між кроками конвеєра;

– перетворення компонентів обробки, які беруть один або декілька об’єктів DataProxy і повертають новий DataProxy. Концепція подібна до перетворень тензорного потоку [33];

– введення даних використовується для читання даних із зовнішніх джерел. На відміну від перетворень, вони не приймають DataProxy як вхідні дані, але створюють вихідні дані DataProxy. Найпоширенішим випадком використання є читання вихідних аудіофайлів і транскриптів;

– виведення даних використовується для експорту результатів обробки у певному форматі. На відміну від перетворень, вони приймають DataProxy як вхідні дані, але не створюють вихідні дані DataProxy. Найпоширеніший сценарій використання – збереження набору даних у форматі певного інструментарію для навчання моделей (наприклад, створення вхідного набору даних для Kaldi або Wav2Vec). Цей компонент дозволяє проводити експерименти з різними архітектурами моделей та інструментами навчання, використовуючи той самий конвеєр попередньої обробки.

Рис. 2.4. Структурна схема трансформера

Комбінація визначених компонентів дозволяє нам визначити конвеєри. Наступним кроком ми визначаємо структури даних та API для них, щоб досягти інтероперабельності між компонентами.

Перш ніж визначати структури даних для компонентів, ми повинні вирішити, як з’єднати окремі компоненти в єдиний конвеєр. Як можна побачити в інших обчислювальних фреймворках [33–35], одним з поширених методів створення складних обчислень з декількох компонентів є використання спрямованих ациклічних графів. Він також підходить для конвеєрів ASR, крім того, використання концепцій, які можуть бути знайомі людям, також зменшує час навчання. Тому ми також представимо структуру обчислень у вигляді орієнтованого ациклічного графа, як показано на рис. 2.5.

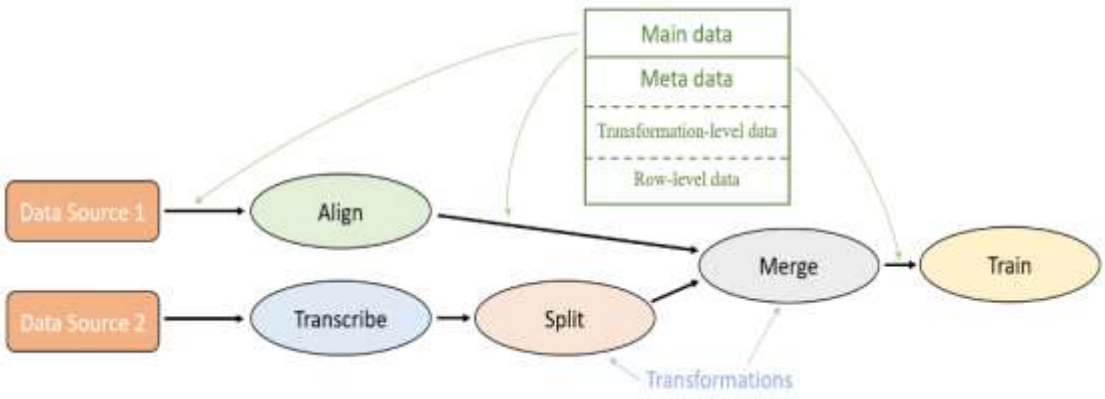


Рис. 2.5. Запропонована структура даних трансформера

Щоб зробити різні компоненти сумісними один з одним, ми повинні визначити формат даних, які будуть передаватися між ними. Структура даних включатиме два основні компоненти: основні дані (наприклад, стенограми та посилання на аудіофайли); метадані (наприклад, інформація про кешування). Метадані можуть складатися з двох частин. Одна частина – це дані на рівні перетворення (наприклад, коли почалося/закінчилося перетворення). Друга частина метаданих – на рівні рядків. Вона зберігатиме метаінформацію для кожного аудіофайлу або висловлювання. Дані на рівні рядків потрібні для підтримки часткової недійсності.

Як базову структуру Suggest використовує Pandas DataFrame. Він має дві особливості, корисні для створення конвеєра ASR:

1. Вбудована підтримка операцій на рівні таблиць, що дозволяє писати більш стислий і чистий код.

2. Існують фреймворки для розподілу обчислень DataFrame, такі як Modin [36], на випадок, якщо в майбутньому ми вирішимо обробляти більші набори даних.

Рекомендований набір стовпців для DataFrame:

– адреса аудіофайлу, як і в першій версії, ми підтримуємо лише локальний шлях до файлу;

– час початку висловлювання визначає момент початку висловлювання. Якщо значення не вказано, ми вважаємо, що у файлі є одне висловлювання;

– час завершення висловлювання визначає момент завершення висловлювання. Якщо значення не вказано – ми вважаємо, що файл містить одне висловлювання;

– текст (транскрипція висловлювання) можна не задавати, якщо подальші перетворення додадуть його;

– тег – це необов’язкова мітка даних, наприклад, його можна використовувати для встановлення джерела даних, щоб мати змогу відстежувати дані в конвеєрі;

– підмножина визначає, чи належить певний запис до навчальної множини, тестової множини або множини валідації. Іншим варіантом може бути наявність двох різних об’єктів. Один для тренувального набору і один для тестового набору. Але оскільки вся попередня обробка однакова для обох, наявність їх в одному фреймі даних спрощує код. Крім того, наявність стовпця з самого початку дає гнучкість у розділенні даних. Всі перетворення проходять через стовпці, які вони не змінюють. Це дозволяє створювати трансформації з однією відповідальністю, які не порушують інші частини даних. Наприклад, ми можемо мати невелику підмножину високоякісних даних для використання в якості набору для перевірки. Тоді стовпець буде створено з компонентів «Введення даних». Або ж ми можемо використати випадковий розподіл як одне з перетворень.

Уніфікація реалізацій перетворень дає наступні переваги:

1. Скорочує час навчання, оскільки користувач лише один раз дізнається, як викликати кожне перетворення та як комбінувати декілька перетворень.

2. Дозволяє мати стандартну реалізацію кешування, яку можна повторно використовувати у різних перетвореннях.

3. Робить компоненти більш стійкими до майбутніх змін, оскільки ми можемо додати глобальний планувальник завдань для розподілених обчислень без переписування перетворень з нуля.

Ми розглянули три способи структурування трансформацій: успадкування, стратегія та шаблони проєктування декораторів. За допомогою успадкування ми можемо реалізувати шаблон проєктування «Шаблонний метод». Спрощена структура для базового класу виглядатиме так:

```
class TransformationBase(ABC):
    @abstractmethod
    def _process(self, inp, **kwargs):
```

```
        # would be overridden in transformations
        pass
    def process(self, inp, **kwargs):
        # validation and caching can go here
        result = self._process(inp, **kwargs)
        # updating cache can go here
        return result
```

Тоді перетворення реалізують метод *_process*. Але такий підхід не дозволить використовувати автозавершення IDE та перевірку типу для параметрів методу «process». Шаблон проєктування стратегій також не дає можливості використовувати автозавершення IDE. Здебільшого він використовується для зміни поведінки об'єкта в деяких аспектах, але у випадку трансформацій ми змінюємо основну частину поведінки, тому це може ввести в оману користувачів. Шаблон проєктування «Декоратор» дозволяє нам покращити поведінку трансформацій, додавши загальну функціональність, таку як кешування, ведення журналу, вимірювання часу, перевірка графіків тощо. Реалізація трансформації за допомогою декоратора може виглядати так:

```
@transformation
class MergeStreams:
    def process(self, data_frames: list[pd.DataFrame]):
        res = pd.concat(data_frames).reset_index()
        log.info("Merged %d sources, got %d records.",
                 len(data_frames), len(res))
        return res
```

Наявність декоратора і графа обчислень дає додаткову перевагу – можемо перевірити, чи використовуються результати конкретного перетворення, і обчислювати тільки те, що потрібно. Нижче наведено приклад конвеєра, який використовує перетворення, визначені та доповнені декораторами:

```
data = MergeStreams().process([
    get_audio_books_stream(),
    get_calls_stream(), ])
data = NormalizeNumbers().process(data)
```

```
data = TrainTestSplit().process(data)
KaldiOutput(dst_dir=data_dst,
        sample_rate=16000).process(data).compute()
```

Метод compute в кінці коду явно запускає обчислення, оскільки він є останнім елементом конвеєра.

Автоматичне кешування та валідація дозволяють прискорити експерименти, переробляти вихідні дані з деякими додатковими кроками тощо. Кешування включає в себе наступні рішення: які перетворення можна і потрібно кешувати; як перевірити, що дані не були змінені; як відстежити, яка частина даних була змінена, щоб мати можливість зробити часткову ануляцію; як зберігати дані. Нижче ми розглянемо ці рішення.

Розглядаючи цільову область (робота з користувацькими наборами даних), найпоширенішим сценарієм буде додавання або видалення частини даних. У таких випадках процедура кешування повинна виявляти змінену частину даних і виконувати обчислення лише над зміненою частиною. Щоб мати змогу робити це на всіх етапах конвеєра, нам потрібен спосіб відображення рядка *DataInput* до рядка або набору рядків у вихідних даних наступних рядків. Візьмемо конвеєр, що складається з наступних кроків:

1. Вхідні дані з *.csv та *.mp3.
2. Нормалізувати числа.
3. Розділіть довгі сегменти на коротші.
4. Перетасуйте і зробіть тренування/тестовий спліт.
5. Збережіть у форматі навчальної структури ASR.

У даному конвеєрі видалення одного вихідного файлу призведе до видалення одного рядка на кроці 2, видалення одного або декількох рядків на кроці 3, видалення тієї ж кількості рядків, але в різних позиціях на кроці 4, іншого результату на кроці 5. Для того, щоб мати можливість відстежувати рядки під час декількох перетворень, нам потрібен спосіб ідентифікувати кожен рядок. Оскільки у нас немає унікальних ідентифікаторів, ми пропонуємо використовувати хеш усіх стовпців DataFrame як ідентифікатор рядка для трансформації.

Як наслідок, для підтримки валідації кешу, у тому числі часткової, список стовпців DataFrame, буде розширено на два стовпці: hash, parent_hash. На вихідному рівні ми зберігаємо три значення хешу: хеш параметрів *__init__;* хеш параметрів процесу; хеш аудіофайлів.

2.3.3. Реалізація алгоритму конвеєра

Оскільки ми намагалися створити конвеєр і використовувати його з реальними завданнями, ми обрали українську мову як мову-кандидата для застосування створеного конвеєра. Оскільки у відкритому доступі є занадто мало наборів даних (267 год.), щоб досягти хороших результатів моделі генерації тексту із мовлення (від англ. Speech-to-Text, STT) без збору та побудови навчального набору даних. У табл. 2.4 представлено загальнодоступні набори даних STT для української мови (мова з низьким рівнем ресурсів), оскільки збір таких загальнодоступних наборів даних є відправною точкою для кожного тренування моделі STT. Такої кількості даних недостатньо для якісної моделі STT. Застосовуючи поточний пайплайн, нам вдалося створити 2 500 год. тренувального набору даних ASR протягом 84 год., що дозволило нам отримати сучасний WER 5,24 для української мови на основі використання Mozilla Common Voice (22 год.) як валідаційного набору даних.

Дані:

– розмічені дані $L = \{x_i, y_i\}_{i=1}^{l}$;

– напіврозмічені дані S;

– нерозмічені дані $U = \{x'_j\}_{j=1}^{u}$.

1. Зібрати загальнодоступні набори даних ASR для поточної мови L.

2. Зібрати загальнодоступні напіврозмічені дані S (аудіозаписи без текстів, наприклад, книги, відеозаписи з транскриптами).

3. Зібрати загальнодоступні нерозмічені дані U (аудіозаписи без текстів, наприклад, громадське радіо).

Таблиця 2.4

Набори даних та джерела для українського мовного корпусу

Набор даних	Клас	Час, год.	Спікери	Якість	Загально-доступний
Український корпус для мовлення [37, 38]	Набір даних ASR	366	330	–	Ні
Багатоголосий корпус «UkReco» [39]	Набір даних ASR	–	<100	–	Ні
Український корпус M-AILABS [40]	Книги	87	6	Висока	Так
Міністерство освіти, культури і науки. Уроки	Youtube	29	<100	Середня	Умови Youtube
Deutsche Welle українською мовою	Youtube	70	<1000	Достатня	Умови Youtube
Телебачення Торонто	Youtube	60	<100	Достатня	Умови Youtube
Mozilla Common Voice	Mozilla	22	235	Висока	Так
TEDx Talks	TEDx	<50	<20	Середня	Умови TED

4. Очистити зібраний набір даних S, застосувавши загальні для ASR етапи підготовки набору даних: виявлення голосової активності, ідентифікацію дедуплікації фрагментів, розширений сегмент мови/музики/шуму та фільтрацію сегментів середнього балу думки.

5. Нормалізувати цифр та чисел для набору даних S.

6. Вирівняти транскрипції аудіо до тексту для набору даних S. Методи вирівнювання описано у [32].

Результат. Акустична та мовна модель p_θ , навчена на наборах даних L та S: ініціалізуйте p_θ , навчивши її лише на розмічених даних L та S.

повторити

1. Розпізнавання нової частини нерозмічених даних $\widetilde{U} \in U$ за 200 год.

2. Повторне розпізнавання раніше використаних нерозмічених даних $\widetilde{U} \in U$ (накопичено).

3. Генератор наборів (фрагментів) даних для навчання.

4. Навчіть p_i акустичні та мовні моделі, використовуючи новий набір даних.

до досягнення збіжності, цільового розміру набору даних, цільового WER або максимальної кількості ітерацій [41].

2.3.4. Критерії оцінки роботи алгоритму розпізнавання природної мови

Оскільки основною метою пайплайну є створення набору даних ASR, ми вирішили не використовувати WER як перший критерій. Перша причина такого рішення полягає в тому, що WER для ASR-моделі, навченої на конкретному наборі даних (або на мові з низькими ресурсами), не можна порівняти з WER моделі, навченої на широкому наборі даних. Друга причина – WER більше залежить від вхідних необроблених аудіоданих (включаючи лексику і словниковий запас в аудіо), ніж від самого програмного пайплайну (неправильний вибір вхідних аудіоданих призведе до поганого WER, навіть якщо програмний пайплайну ідеально створить навчальний набір даних (наприклад, вхідні аудіо дебатів, але виробнича область – медицина).

Таким чином, було використано наступні критерії для вимірювання результатів:

1. Основними критеріями є здатність пайплайну створювати відповідний набір даних ASR з більш ніж 2 500 год. вихідних даних, навчати ASR з використанням існуючих фреймворків.

2. Другим критерієм є час, необхідний для створення навчального набору даних у 2 500 год. Людині знадобиться щонайменше стільки ж часу для маркування даних (щонайменше 2 500 год. для прослуховування кожного вхідного аудіосегмента). Отже, ми встановили верхню межу в 2 500 год. обчислювального часу, використовуючи загальнодоступний набір даних на GPU довжиною 2 500 год.

Як третій критерій, ми виміряли валідаційний WER, щоб зрозуміти збіжність навчання. Для вимірювання WER навченої моделі ми використовували набір даних Mozilla Common Voice 22 год. українською мовою.

Як ми бачимо з реалізованого прикладу, є багато незалежних кроків і більш важливих повторюваних кроків, які займають багато часу навіть у досвідчених інженерів. Застосовуючи основні будівельні блоки конвеєра до кожного кроку: DataProxy, Перетворення, Введення та Виведення даних, ми змогли обробити від 267 год. ASR-даних до 2 500 год. ASR-даних всього за 1289 год., використовуючи 4×1080ti GPU та 1 CPU AMD 3960x. Крім того, автоматичне кешування та валідація дозволяє прискорити такі експерименти та ітерації, переробку вихідних даних з деякими додатковими кроками тощо. Наприклад, хешування 10 000 год. аудіофайлів зайняло 25 хв. у разі використання SSD-накопичувача. Використання HDD призвело б до ще більшої тривалості перевірки хешу. Іншим варіантом є використання часу модифікації файлів. Використання часу модифікації є менш надійним у деяких випадках, наприклад, він може пропустити заміну файлу на старішу версію файлу. Оскільки однією з головних цілей є низька крива навчання, ми вирішили залишити хеші за замовчуванням і додати можливість заміни стратегії перевірки файлів як частину конфігурації фреймворку.

За допомогою поточного пайплайну ми змогли досягти найсучаснішого WER 5.24 для української мови, і ми очікуємо, що аналогічного WER можна досягти майже для будь-якої мови, яка має принаймні 250 год. даних як відправну точку для використання поточного пайплайну.

Розроблено та реалізували програмний конвеєр ASR для навчання підходу до формування наборів даних з необроблених аудіозаписів (для потрібної мови або специфічних для домену нерозмічених аудіозаписів).

Основними цілями такого пайплайну були зменшення витрат часу дослідника або розробника на підготовку навчального набору даних потрібного розміру та скорочення навчального процесу для новачків, які починають працювати з доступними фреймворками та інструментами ASR.

2.4. Способи підвищення ефективності розпізнавання мовної інформації

2.4.1. Розпізнавання багатомовних мовленнєвих емоцій

Досягнення в галузі автоматизованого розпізнавання мови значно прискорили автоматизацію контакт-центрів [42, 43]. Така автоматизація та доповнення людських агентів вимагає перекладу мовлення в текст, щоб зрозуміти, що було сказано, і його сенс [44]. Аналіз настроїв на основі тексту досить часто не може розпізнати гнів і щастя, якщо людина не виражає або не артикулює їх за допомогою певних слів. У той же час люди артикулюють/кодують емоції в інтонаціях [45, 46]. Це створює потребу в надійному розпізнавання мовних емоцій (від англ. Speech Emotion Recognition, SER) як невід'ємній частині вимірювання показника чистого промоутерського балу.

Через відсутність розмічених наборів даних для розпізнавання емоцій та низьку переносимість моделей, навчених суто на англійській акустиці, невеликі набори емоційних даних для інших мов [47–53]. Незважаючи на це, лише англійська та китайська мови представлені багатьма наборами даних і визначають різноманітну акустику, лексику тощо. Більшість одномовних наборів даних є відносно невеликими, що недостатньо для перекладу навіть на мови однієї групи. Проблема збору емоційних даних шляхом відокремлення від реальних випадків і маркування дзвінків є болючим питанням і здається малоймовірною. Тому основний дослідницький інтерес змістився на створення багатомовних моделей SER [54, 55].

У [56] досліджуються можливості точного налаштування попередньо навченої моделі SER з невеликою кількістю даних, що дає багатообіцяючі результати, але все ще вимагає ручної підготовки набору даних для цільової мови. В [57] запропоновано об'єднати акустичні особливості в тришарову перцептивну модель емоцій, яка показує результати, порівнянні з одномовною SER на новій мові без навчання. А в [58] запропоновано ансамблеве навчання, яке також дає

багатообіцяючі результати. Загалом, збірка та нашарування виглядають вигідними, але не доступні в основних фреймворках та інструментах розпізнавання мови «з коробки».

У [59] та [60] запропоновано двопрохідну схему класифікації, що складається з ідентифікації природної мови та SER, що також обмежує розгортання у виробництві наявністю моделей SER для ідентифікованих мов. У [61] запропоновано неконтрольовані підходи, які можуть показувати чудові результати, але все ще вимагають багато обчислювальних ресурсів.

Класифікація емоцій та оцінка емоційної залученості є одними з головних потреб кожного бізнесу. Як реагують клієнти, що викликає радість, а що – смуток – нескінченні питання для багатьох компаній. Ми прагнемо виявити можливість надійного багатомовного застосування розпізнавання емоцій через оцінку міжмовного розпізнавання емоцій.

Основним обмеженням для практичної реалізації SER є наявність наборів даних для кожної конкретної мови, оскільки схожість фонем у різних мовах сильно відрізняється. Важко створити набір даних для кожної мови для завдання SER, і тому моделі навчаються на одній мові, а реалізуються по-різному.

Найпоширеніший спосіб і набір даних для розпізнавання емоцій – мультимодальний. Особливо ефективними для розпізнавання емоцій є відеодані, що є рідкісним випадком для виробничого середовища, наприклад, контакт-центру. У такому середовищі для прогнозування емоцій доступна лише аудіоінформація.

Емоційні процеси корелюють з акустичними параметрами (частота, спектральна енергія, швидкість мовлення, мерехтіння тощо). Крім того, хороші результати SER демонструють частотні цепстральні коефіцієнти Мела, спектральний зсув, характеристики енергетичного оператора Тігера, спектрограми та особливості глоткової форми хвилі [62].

Оскільки дослідження в галузі пошуку найкращого способу представлення акустичних сигналів і вилучення особливостей рухів значно просунулися вперед, найбільш інтригуючим є питання, де знайти релевантні дані для навчання SER-моделей.

Хоча для більшості просунутих мов у галузі NLP/мовлення не є проблемою знайти більше 200 год. розмічених даних, для більшості мов це все ще довгий шлях. У цій ситуації найочевиднішим підходом є навчання моделі на мові з багатим набором даних і застосування її у виробничому середовищі з локальною мовою. Звучить добре, але на практиці це працює далеко не так добре. Ми маємо на меті оцінити переносимість моделей на різні мови як підхід і з'ясувати, на що слід звернути увагу при цьому.

2.4.2. Підвищення точності розпізнавання природної мови для близькоспоріднених мов

Здатність точно ідентифікувати природні мови лежить в основі численних додатків, починаючи від автоматизованих систем маршрутизації дзвінків, багатомовних голосових помічників і закінчуючи аналітичним програмним забезпеченням для багатомовних агентів колл-центрів [63]. Завдяки швидкому прогресу в галузі DL, системи переклад мови жестів (від англ. Sign Language Interpreting, SLI) досягли надзвичайної точності на різноманітних наборах даних [64].

Однак, як і у випадку з багатьма технологіями, які обіцяють універсальне застосування, існує застереження. Не всі мови ідентифікуються з однаковою точністю. Хоча розрізнення акустично відмінних мов, таких як англійська та мандаринська [65], може бути простим для більшості моделей, справжній виклик виникає, коли завдання включає мови з багатьма фонетичними, історичними та акустичними характеристиками. Розглянемо, наприклад, скандинавські мови норвезьку та шведську або південноазійські мови урду [66] та гінді [67]. Через свою переплетену історію та спільне лінгвістичне коріння ці мови створюють значні проблеми для звичайних систем SLI. Помилкові ідентифікації можуть бути частими, а наслідки можуть варіюватися від незначних незручностей у

користувацьких програмах до значних непорозумінь у більш критичних сценаріях [68].

Важливо підкреслити, що хоча багато моделей SLI можуть похвалитися високими показниками точності, ці цифри часто можуть приховувати нюанси. При подальшому аналізі ми помічаємо, що хоча ці моделі демонструють винятково хороші результати на різних мовах, їхня продуктивність може значно погіршитися, коли вони стикаються з акустично подібними мовами [69, 70]. Такі відкриття вимагають глибшого занурення у проблеми схожих мов і потенційні рішення. Ця стаття має на меті дослідити вплив даних на точність моделей SLI, зосереджуючись на порівнянні моделей, навчених на різних наборах даних, порівнянні якості та кількості даних, а також на вимогах до балансу даних для досягнення найкращої моделі SLI.

Окресливши виклики, пов’язані з розрізненням близькоспоріднених мов, ми підготували ґрунт для впровадження нової стратегії тонкого налаштування. За допомогою цього підходу ми прагнемо підвищити дискримінаційну здатність систем SLI, дозволяючи їм розрізняти навіть найтонші нюанси між схожими мовами, тим самим розширюючи межі досяжного в ідентифікації природної мови [71].

Шен та ін. [72] провели великий огляд сучасної літератури про SLI, приділивши особливу увагу особливостям мовлення та архітектурним рішенням. Що стосується архітектури, то старі моделі SVM і HMM для таких завдань сильно програють новим моделям. Наприклад, [73] та [74] досягли точності в середньому 66,55% для кожної мови (навчені лише для чотирьох мов), що набагато нижче, ніж будь-яка сучасна архітектура, навіть навчена для 107 мов.

Драгічі та ін. [75] дослідили попередні проєкти систем SLI, зокрема ті, що використовували архітектури CNN та згорткових рекурентних нейронних мереж (CRNN). Їхні дослідження також включали набір з семи мов. Незважаючи на зростаючу складність цих систем, вони досягли помітного успіху: архітектура CNN досягла точності 71%, а архітектура CRNN – 83%.

Саме тому для розв’язання задач SLI використовуються сучасні архітектури нейронних мереж з акцентованою увагою до каналів, поширенням та агрегацією в нейронних мережах із затримкою в часі (ECAPA-TDNN) [20, 76, 77], CNN [78, 79] та міжмовних мовних репрезентацій (XLSR) [80, 81], які значно перевершують старі архітектури на основі LSTM. У статті [82] оцінюються дві нові моделі архітектур TC-ResNet10 і LECAPAT, зосереджені на швидкості та досягненні близьких до великих архітектур результатів точності, але майже на два порядки швидших і на чотири порядки менших. Хоча швидкість неймовірна, для бізнес-задач все ще неприйнятні компроміси з точки зору точності.

Що стосується доступних наборів даних, то найбільш широко визнані стандарти для оцінки нових моделей і технологій SLI / розпізнавання мови жестів (від англ. Sign Language Recognition, SLR) базуються на наборах даних NIST LRE [83]. Ці набори даних переважно складаються з вузькосмугового розмовного телефонного мовлення. Значні обсяги розмовних телефонних даних з конкретних мов використовуються для розробки конкурентоспроможних систем для NIST LRE. Консорціум лінгвістичних даних (від англ. Linguistic Data Consortium, LDC) часто надає такі дані, але вони можуть бути дорогими і обчислюватися тисячами доларів. Наприклад, створений фреймворк Kaldi [10] для LRE07 вимагає 18 різних наборів даних LDC SLR, що загалом коштує 15 400 доларів США для країн, які не є членами LDC3. Така висока вартість є значним бар’єром для нових дослідницьких груп, які прагнуть зробити свій внесок в академічну галузь SLR.

З відкритих наборів даних найвідомішим є VoxLingua107 [84], який є чудовим і відкрив двері для навчання багатомовних моделей SLI замість 5–10 мов, оскільки він складається з 107 мов. З іншого боку, обмеження поточного набору даних полягають у тому, що для VoxLingua107 набір для розробки з перевіреними вручну мовними мітками є дуже обмеженим і містить мітки лише для 33 мов, а через автоматизований процес збору даних точність набору даних для навчання для близькоспоріднених мов є низькою для деяких мов (наприклад, багато іспанських діалогів у галісійському наборі або багато російської мови в українському наборі).

Набір даних Common Voice (CV) [85] також має дуже обмежену кількість мов. Удосконалення моделі SLI описано в [86], де основна увага приділяється впливу іноземних акцентів, і [87], де основна увага приділяється ефективності моделі та класифікації невидимих мов і різних акустичних середовищ без додаткового навчання.

2.5. Обмеження та ризики використання методів розпізнавання голосової інформації в системах кібербезпеки

2.5.1. Переваги застосування методів розпізнавання голосової інформації

Технології розпізнавання голосу та NLP значно *підвищують ефективність* роботи з великими обсягами даних. Вони дозволяють швидко обробляти текстову та голосову інформацію, виконуючи це набагато швидше, ніж люди. Крім того, автоматизація рутинних завдань, таких як транскрибування дзвінків, сортування електронної пошти або аналіз звітів, сприяє значному скороченню часу, витраченого на повторювані операції.

Завдяки автоматичному аналізу комунікацій технології можуть ефективно розпізнавати загрози, такі як терористичні змови або кіберзагрози. Це дозволяє значно *підвищити рівень безпеки*. Крім того, біометрична аутентифікація на основі голосу забезпечує надійний захист від несанкціонованого доступу, що є важливим аспектом сучасної кібербезпеки.

Голосові помічники та чат-боти відіграють важливу роль у *покращенні якості обслуговування* користувачів. Вони забезпечують швидку та ефективну допомогу, знижуючи навантаження на операторів. Також NLP інтерфейси полегшують взаємодію з технологіями для людей, які стикаються з труднощами при використанні традиційних інтерфейсів.

Розпізнавання голосу дозволяє трансформувати голосові повідомлення у текст для подальшого аналізу за допомогою NLP. Це знаходить своє застосування,

наприклад, у записах дзвінків у *службах підтримки клієнтів* (в чому числі інтерактивна голосова відповідь). Аналіз тексту на основі NLP допомагає виявляти ключові слова, настрої та тенденції, що особливо корисно для аналізу соціальних мереж та вивчення громадської думки. Чат-боти та голосові помічники автоматизують процес спілкування з користувачами, забезпечуючи відповіді на запити та надання необхідної інформації. Серед прикладів сучасних технологій можна відзначити Google Assistant, Amazon Alexa та Apple Siri, які використовують розпізнавання голосу та NLP для взаємодії з користувачами. IBM Watson надає інструменти для аналізу тексту та розпізнавання голосу у бізнес-додатках, а Microsoft Azure Cognitive Services пропонує набір сервісів для розробки додатків із можливостями NLP та розпізнавання мовлення.

Також важливо відзначити NLP моделі, такі як Google BERT, яка використовує трансформери для розуміння контексту слів у реченнях. OpenAI GPT є генеративною моделлю для NLP, здатною генерувати тексти та виконувати інші завдання NLP. Mozilla DeepSpeech пропонує відкриту модель для розпізнавання мовлення на основі нейронних мереж, що також є важливим інструментом у цій галузі.

Слід також зазначити фактори, що впливають на вартість кінцевої системи:

1. Інфраструктура (сервери, хмарних сервіси, обчислювальні потужності).
2. Ліцензії та програмне забезпечення (комерційні моделі та платформи).
3. Розробка, налаштування та інтеграцію технологій в існуючі системи.
4. Персонал (залучення, навчання та зарплата).
5. Безпека (програмне та апаратне забезпечення).

Зазначені вище фактори безпосередньо впливають на вартість проєктів, яка наведена в табл. 2.5. Особливо великий вплив мають такі фактори як інфраструктура, розробка та налаштування, а також заходи безпеки, що є критичними для різних масштабів впровадження технологій. Наведено оцінку вартості проєктів різного розміру, пов’язаних із впровадженням технологій розпізнавання мовлення та NLP. Таблиця демонструє три категорії проєктів: малий, середній та великий, кожен з яких відрізняється сферою застосування та

орієнтовною вартістю реалізації. Вартість варіюється від декількох тисяч доларів для простих чат-ботів або голосових помічників до кількох мільйонів для масштабних систем, впроваджених у державних або великих комерційних структурах.

Таблиця 2.5

Оцінка вартості проєктів різного розміру

Розмір проєкту	Сфера застосування	Орієнтовна вартість, тис. дол.
Малий	Впровадження простих чат-ботів або голосових помічників	Одиниці – десятки
Середній	Розробка та інтеграція систем розпізнавання мовлення та NLP для бізнес-цілей	Десятки – сотні
Великий	Впровадження масштабних систем для державних чи великих комерційних структур	Тисячі

Розпізнавання голосу та NLP є потужними інструментами, які можуть значно підвищити ефективність і безпеку держави. Однак їх впровадження потребує значних інвестицій та ретельного планування для забезпечення конфіденційності, надійності та безпеки даних. Поширені моделі, такі як BERT та GPT, надають великі можливості для аналізу та розуміння тексту, а сучасні імплементації показують високу ефективність цих технологій у різних сферах.

2.5.2. Обмеження реалізацій методів розпізнавання голосової інформації

Основними обмеженнями систем є в першу чергу є вузкість застосування і по друге швидкість адаптації новітніх технологій. З розвитком технологій і особливо з появою новітніх методів генерації тексту та синтезу голосу, кіберзлочинці отримали нові інструменти для здійснення більш складних та ефективних атак. Однією з найбільш небезпечних сучасних загроз є атаки, пов’язані з видаванням себе за іншу людину, також відомі як спуфінг або імперсонація.

Новітні *моделі клонування голосу* на основі DL, такі як Deepfake Voice або інші генеративні моделі, дозволяють створювати високоточні копії голосу реальних людей. Ці моделі здатні відтворювати інтонації, манери мовлення і навіть емоції. За наявності достатньої кількості аудіозаписів голосу людини зловмисники можуть створити голосовий клон. Такі технології використовуються для атак на високопоставлених осіб або бізнес-лідерів, коли зловмисники видають себе за них у телефонних розмовах або через голосові повідомлення. Це може призвести до виконання неправомірних фінансових транзакцій, отримання конфіденційної інформації або скомпрометування корпоративної безпеки. Наприклад, відомі випадки, коли компанії зазнавали значних збитків через фальшиві дзвінки від імені керівників, які «наказували» перевести гроші на рахунки хакерів.

Поєднання технологій клонування голосу і *методів соціальної інженерії* робить атаки ще більш ефективними. Соціальна інженерія базується на психологічних маніпуляціях, коли жертва вводиться в оману з метою виконання певних дій. Використовуючи згенеровані голоси, зловмисники можуть переконати жертв у тому, що вони спілкуються з відомою їм особою, що підвищує ймовірність успішної атаки. Ці атаки можуть націлюватися не лише на фінансові транзакції, а й на доступ до конфіденційної інформації, зокрема паролів, внутрішніх документів або систем безпеки. Вони також можуть використовуватися для дезінформації та поширення фейкових новин, що створює загрози на рівні державної безпеки.

З розвитком *технологій біометричної аутентифікації* все більше систем використовують голос як засіб підтвердження особи. Проте ці системи також можуть бути вразливими до атак з використанням згенерованих голосів. Голосові паролі, які раніше вважалися надійним засобом захисту, тепер можуть бути обмануті за допомогою технологій клонування голосу. Хакери можуть отримати доступ до облікових записів, банківських рахунків або інших критичних ресурсів, використовуючи підроблені голоси. Це ставить під загрозу не лише окремих користувачів, а й цілу інфраструктуру компаній та державних організацій, де голосова аутентифікація використовується для доступу до конфіденційних даних або систем.

Для захисту від таких атак необхідно вживати комплексні заходи:

– підвищення обізнаності персоналу через освітні програми для співробітників та користувачів для вчасного розпізнавання потенційних загроз і повідомлення про підозрілі випадки;

– мультифакторна аутентифікація, включаючи фізичні фактори, такі як токени або біометричні дані інших типів (відбитки пальців, розпізнавання обличчя і голосу);

– використання ШІ для виявлення фальсифікацій для аналізу голосових команд і виявлення аномалій або ознак підробки;

– постійне оновлення систем захисту для протидії новим типам атак.

Для успішної протидії таким загрозам необхідні інноваційні підходи до захисту та постійний розвиток систем кібербезпеки.

2.5.3. Ризики застосування методів розпізнавання голосової інформації

Технології розпізнавання голосу та NLP мають великий потенціал для підвищення ефективності державної безпеки. Однак їх використання вимагає уважного підходу до питань конфіденційності, безпеки даних, надійності, етики та захисту від зловживань. Для запобігання ризикам, пов’язаним із цими технологіями, держави повинні впроваджувати чіткі регуляції та забезпечувати високий рівень захисту.

Ключовими аспектами викликів та ризиків якими супроводжується використання технологій розпізнавання голосу та NLP є:

1. *Надійність та точність*. Технології розпізнавання голосу та NLP можуть допускати помилки, що впливають на прийняття рішень. Неточні результати можуть призвести до неправомірних дій або пропущених загроз. Наприклад, неправильна ідентифікація голосу підозрюваного може призвести до хибних арештів або пропущених терористичних атак.

2. *Наявність даних*. Одним з ключових обмежень технологій розпізнавання голосу є залежність від якості та кількості даних. Для тренування моделей розпізнавання мовлення потрібні великі обсяги даних, які включають різноманітні зразки голосу з різних джерел, мов і діалектів. Недостатність або неповнота таких даних може призвести до зниження точності системи. Також варто враховувати питання різноманітності даних. Технології можуть демонструвати нижчу ефективність при роботі з мовами або діалектами, для яких доступно мало навчальних даних. Наприклад, якщо модель навчена переважно на англомовних зразках, вона може працювати менш ефективно з іншими мовами або з користувачами з акцентами.

3. *Вартість впровадження*. Впровадження технологій розпізнавання голосу є дорогим процесом, що потребує значних фінансових ресурсів. Основними статтями витрат є: інфраструктура (необхідність інвестувати в потужне обладнання або хмарні ресурси для обробки великих обсягів голосових даних), програмне забезпечення та ліцензії (високі витрати на придбання ліцензій для комерційних моделей та платформ, які надають функції розпізнавання голосу та NLP), розробка та інтеграція (витрати на розробку індивідуальних рішень, налаштування моделей під конкретні завдання та їх інтеграцію з існуючими системами), персонал (залучення фахівців з NLP, ML та кібербезпеки, а також їх навчання й підтримка) і кібербезпека (витрати на захист даних та створення системи безпеки, яка зможе протистояти можливим загрозам, таким як злом або витік даних). Усе це робить технології розпізнавання голосу недоступними для малих і середніх підприємств або організацій з обмеженими бюджетами а також країн, що є одним із суттєвих обмежень їх широкого впровадження.

4. *Конфіденційність та приватність*. Збір та аналіз голосових і текстових даних можуть порушувати права на приватність громадян. Неправильне використання цих технологій може призвести до незаконного стеження та втручання в особисте життя. Наприклад, використання державою технологій для моніторингу телефонних розмов без належних правових підстав може порушувати права людини.

5. *Безпека даних*. Збереження та обробка великих обсягів голосових і текстових даних потребують високого рівня захисту. Витік або злом таких даних можуть мати серйозні наслідки для державної безпеки. Зокрема, злом баз даних, які містять розмови високопосадовців, може надати доступ до чутливої інформації ворогам держави.

6. *Використання у військовій та розвідувальній діяльності*. Технології можуть бути використані для збору розвідувальних даних, але водночас можуть стати об'єктом атак з боку противників, які прагнуть дезінформувати або зламати системи. Наприклад, противник може використовувати NLP для створення дезінформаційних кампаній або зламу голосових командних систем.

7. *Етичні питання*. Використання технологій повинно враховувати етичні аспекти, зокрема недопущення дискримінації та забезпечення прозорості у використанні даних. Важливо розробляти алгоритми, які неупереджено ставляться до всіх груп населення, без дискримінації за ознаками раси, статі чи віку.

8. *Підробка голосу* та *штучні голоси*. Розвиток технологій підробки голосу 'deepfake' створює ризики для безпеки, оскільки зловмисники можуть використовувати ці технології для створення фальшивих повідомлень або команд. Наприклад, підробка голосу високопосадовця може призвести до хибних наказів або дезінформації.

2.5.4. Виклики щодо впровадження технологій розпізнавання голосу

Актуальність і новизна технологій розпізнавання голосу та NLP є одними з найважливіших тем у сучасному світі, особливо з огляду на зростання ролі цифрових комунікацій та необхідність забезпечення національної безпеки. Ці технології відіграють ключову роль у багатьох аспектах державного управління, комерційної діяльності та навіть повсякденного життя, оскільки вони дозволяють автоматизувати аналіз великих обсягів інформації, що надходить у вигляді голосових повідомлень або текстових даних. Це стає критично важливим для таких

завдань, як збір розвідувальних даних, моніторинг громадської безпеки та забезпечення правопорядку, а також для запобігання потенційним загрозам.

Актуальність технологій розпізнавання голосу та NLP обумовлена зростаючою роллю цифрових комунікацій у сучасному суспільстві. У світі, де інформація стає основним ресурсом, здатність швидко й точно аналізувати великі обсяги даних має вирішальне значення. Технології NLP та розпізнавання голосу дозволяють значно скоротити час, необхідний для обробки інформації, підвищуючи тим самим ефективність роботи державних та приватних організацій.

Однією з найбільш актуальних сфер застосування цих технологій є національна безпека. Використання NLP для моніторингу комунікацій дозволяє виявляти потенційно небезпечні ситуації на ранніх етапах. Наприклад, аналіз текстових або голосових повідомлень може допомогти у виявленні підозрілої активності, що є важливим у контексті боротьби з тероризмом та іншими загрозами державної безпеки.

У комерційному секторі технології розпізнавання голосу й NLP також знаходять широке застосування. Вони використовуються для підвищення якості обслуговування клієнтів, автоматизації рутинних завдань, таких як сортування електронної пошти або транскрибування дзвінків, а також для покращення маркетингових стратегій через аналіз соціальних мереж і відгуків клієнтів.

Новизна технологій розпізнавання голосу та NLP полягає у швидкому розвитку і вдосконаленні методів ML, які використовуються для підвищення точності та ефективності цих технологій. Однією з найбільш значущих новацій останніх років є розвиток моделей DL, таких як BERT від Google і GPT від OpenAI, які дозволяють значно поліпшити результати NLP.

Ці моделі здатні розуміти контекст слів у реченнях, що дозволяє їм не тільки точніше інтерпретувати текст, але й генерувати нові повідомлення, що виглядають, як створені людиною. Такий прогрес відкриває нові горизонти для автоматизації у сферах, де раніше це було неможливо. Наприклад, сучасні чат-боти і голосові помічники стали настільки «розумними», що можуть вести складні діалоги з користувачами, забезпечуючи їм підтримку практично у будь-яких питаннях.

Крім того, новітні технології дозволяють створювати персоналізовані системи взаємодії з користувачами. Наприклад, технології розпізнавання голосу можуть навчатися на індивідуальних особливостях мовлення конкретної людини, що дозволяє створювати більш точні та ефективні системи, які здатні реагувати на особливі вимоги користувача.

Попри великий потенціал і нові можливості, які надають технології розпізнавання голосу та NLP, їх впровадження супроводжується рядом викликів. Одним з головних викликів є питання конфіденційності та безпеки даних. Збирання та обробка великих обсягів голосових і текстових даних можуть створювати ризики для приватності користувачів, що викликає занепокоєння у громадськості та потребує відповідного законодавчого регулювання.

Ще одним важливим викликом є надійність і точність технологій. Незважаючи на прогрес у галузі, помилки в розпізнаванні голосу або аналізі тексту все ще можливі, що може призвести до серйозних наслідків, особливо у сферах, де точність має критичне значення, таких як державна безпека або охорона здоров’я.

Етичні питання також стають все більш актуальними у зв’язку з розвитком цих технологій. Наприклад, алгоритми для аналізу даних, можуть бути упередженими щодо певних груп населення, що може призводити до дискримінації [21].

Висновки до розділу 2

1. Сучасні практичні публікації щодо обробки природної мови мають значний обсяг накопленого досвіду, тому був проведений детальний аналіз метрик природної мови та критеріїв для оцінювання якості її обробки. Дані методи зосереджені, в першу чергу, на аналізу якості, але не на доступність або цілісність, тому потребують додаткового дослідження процесів нелегального та/або неетичного використання даних технологій.

2. Вперше запропонований та математично обґрунтований метод автоматизованого конвеєру для створення навчальних наборів даних з

нерозмічених аудіозаписів, який при навчанні на невеликій кількості нерозмічених даних дозволяє реалізувати підхід автоматичного отримання високоточного маркування, який дозволяє тренувати мовні моделі при наявності незначного обсягу маркованих аудіоданих (починаючи від 250 год.), що знижує вартість формування тренувального набору даних порівняно з ручним на 84% і пришвидшує процес маркуванням щонайменше на 85%, що в свою чергу знижує вартість тренування моделей на 61% і пришвидшує процес мінімум на 69%.

3. Формалізована модель автоматизованого конвеєру дозволила створити навчальні набори даних з нерозмічених аудіозаписів та визначити критерії оцінки її роботи. Для цього був розроблений програмний код для автоматизованого створення навчальних наборів даних на основі нерозмічених аудіозаписів, що є обмеженням для навчання ASR-моделей для мов з низькими ресурсами та із специфічних доменів.

4. Логічним кроком у дослідженні стало визначення базових способів підвищення ефективності розпізнавання мовної інформації при одночасній роботі із кількома мовами при визначенні емоційного стану суб’єкта. На прикладі конкретних мов було показано, що використання програмного коду з представленими компонентами є достатнім для повної автоматизації створення наборів даних розпізнавання мови на основі сирих нерозмічених аудіозаписів та надання інженерам-програмістам інструменту для створення якісних наборів даних розпізнавання мови, а отже, для навчання моделей для мов з низькими ресурсами та залучення більшої кількості інженерів у сферу розпізнавання мовлення.

5. Адекватність формалізованих переваг, обмежень, ризиків та викликів при впровадженні та застосуванні методів розпізнавання голосової інформації вимагає ретельного підходу. Було визначено, шо держави та компанії повинні зосередитися на розробці чітких регуляцій, які захищатимуть права громадян і забезпечуватимуть надійність та етичність використання мовних технологій.

Список використаних джерел у розділі 2

1. Марценюк, М., Козачок, В., Богданов, О., Іосіфов, Є., & Бржевська, З. (2023). Аналіз методів виявлення дезінформації в соціальних мережах за допомогою машинного навчання. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2(22), 148–155. https://doi.org/10.28925/2663-4023.2023.22.148155

2. Dasgupta, S., Piplai, A., Kotal, A., & Joshi, A. (2020). A Comparative Study of Deep Learning based Named Entity Recognition Algorithms for Cybersecurity. In *2020 IEEE International Conference on Big Data* (Vol. 9, pp. 2596–2604). IEEE. https://doi.org/10.1109/bigdata50022.2020.9378482

3. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Kipchuk, F., & Sukaylo, I. (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 83, pp. 25–36). https://doi.org/10.1007/978-3-030-80472-5_3

4. Іосіфов, Є., & Соколов, В. (2024). Методи аналізу природної мови та застосування нейронних мереж в кібербезпеці. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 4(24), 398–414. https://doi.org/10.28925/2663-4023.2024.24.398414

5. Poulter, C. (2020). Voice Recognition Software – Nuance Dragon Naturally Speaking. In *Occupational Medicine* (Vol. 70, no. 1, pp. 75–76). Oxford University Press (OUP). https://doi.org/10.1093/occmed/kqz128

6. Wang, H. H. (2021). Speech Recorder and Translator using Google Cloud Speech-to-Text and Translation. In *Journal of IT in Asia* (Vol. 9, no. 1, pp. 11–28). UNIMAS Publisher. https://doi.org/10.33736/jita.2815.2021

7. The Cloud and Microsoft Azure Fundamentals. (2019). In *Microsoft Azure Infrastructure Services for Architects* (pp. 1–46). Wiley. https://doi.org/10.1002/9781119596608.ch1

8. Leeper, T. J. (2018). aws.transcribe: Client for “AWS Transcribe” [Dataset]. In CRAN: Contributed Packages. In *The R Foundation*. https://doi.org/10.32614/cran.package.aws.transcribe

9. Pickering, J. (2024). Cosegmentation in the IBM Text-to-Speech System. In Speech and Hearing. In *Autumn Conference 1986*. Institute of Acoustics. https://doi.org/10.25144/22372

10. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. K., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., & Veselý, K. (2011). The Kaldi Speech Recognition Toolkit. In *Proceedings of the ASRU* (pp. 1–4).

11. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A. Y. (2014). Deep Speech: Scaling up End-to-End Speech Recognition (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.1412.5567

12. Lee, A., & Kawahara, T. (2009). Recent Development of Open-Source Speech Recognition Engine Julius. In *Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference* (pp. 131–137).

13. Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnicky, A. I. (2006). Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings* (Vol. 1, pp. I-185–I-188). IEEE. https://doi.org/10.1109/icassp.2006.1659988

14. Recognition of Citizens’ Voice with Social Media. (2019). SAGE Publications Ltd. https://doi.org/10.4135/9781526486882

15. Agnitio Launches Voice Authentication for Android. (2012). In *Biometric Technology Today* (Vol. 2012, no. 5, p. 12). Mark Allen Group. https://doi.org/10.1016/s0969-4765(12)70094-2

16. Beyond the Standard Model of Verbal Probing. (2005). In *Cognitive Interviewing* (pp. 87–101). SAGE Publications, Inc. https://doi.org/10.4135/9781412983655.n6

17. Kulke, L., Feyerabend, D., & Schacht, A. (2020). A Comparison of the Affectiva iMotions Facial Expression Analysis Software with EMG for Identifying Facial Expressions of Emotion. In *Frontiers in Psychology* (Vol. 11). Frontiers Media SA. https://doi.org/10.3389/fpsyg.2020.00329

18. Vocapia Research SAS. (2024). VoxSigma Speech to Text Software Suite. https://www.vocapia.com/voxsigma-speech-totext.html

19. Ash, T., Francis, R., & Williams, W. (2018). The Speechmatics Parallel Corpus Filtering System for WMT18. In *Proceedings of the 3rd Conference on Machine Translation: Shared Task Papers* (pp. 853–859). https://doi.org/10.18653/v1/w18-6472

20. Iosifov, I., Iosifova, O., Romanovskyi, O., Sokolov, V., & Sukailo, I. (2022). Transferability Evaluation of Speech Emotion Recognition Between Different Languages. *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 134, pp. 413–426). https://doi.org/10.1007/978-3-031-04812-8_35

21. Іосіфов, Є., & Соколов, В. (2024). Порівняльний аналіз методів, технологій, сервісів та платформ для розпізнавання голосової інформації в системах забезпечення інформаційної безпеки. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 1(25), 468–486. https://doi.org/10.28925/2663-4023.2024.25.468486

22. Derczynski, L. (2016). Complementarity, F-Score, and NLP Evaluation. In *Proceedings of the 10th International Conference on Language Resources and Evaluation* (pp. 261–266).

23. Klakow, D., & Peters, J. (2002). Testing the Correlation of Word Error Rate and Perplexity. In *Speech Communication* (Vol. 38, no. 1–2, pp. 19–28). https://doi.org/10.1016/s0167-6393(01)00041-3

24. Santosh, D. T. (2018). A Combined Approach for Effective Features Extraction from Online Product Reviews. In *International Journal of Education and Management Engineering* (Vol. 8, no. 1, pp. 11–21). MECS Publisher. https://doi.org/10.5815/ijeme.2018.01.02

25. Khodadi, I., & Abadeh, M. S. (2014). A Memetic-Based Approach for Web-based Question Answering. In *International Journal of Information Technology and*

Computer Science (Vol. 6, no. 9, pp. 39–45). MECS Publisher. https://doi.org/10.5815/ijitcs.2014.09.05

26. Protim Ghosh, P., Shahariar, R., & Hossain Khan, M. A. (2018). A Rule based Extractive Text Summarization Technique for Bangla News Documents. In *International Journal of Modern Education and Computer Science* (Vol. 10, no. 12, pp. 44–53). MECS Publisher. https://doi.org/10.5815/ijmecs.2018.12.06

27. Jain, H. (2017). A Web based Application for Sentiment Analysis. In *International Journal of Education and Management Engineering* (Vol. 7, no. 1, pp. 25–35). MECS Publisher. https://doi.org/10.5815/ijeme.2017.01.03

28. Kokare, R., & Wanjale, K. (2015). A Natural Language Query Builder Interface for Structured Databases using Dependency Parsing. In *International Journal of Mathematical Sciences and Computing* (Vol. 1, no. 4, pp. 11–20). MECS Publisher. https://doi.org/10.5815/ijmsc.2015.04.02

29. Bais, H., Machkour, M., & Koutti, L. (2016). A Model of a Generic Natural Language Interface for Querying Database. In *International Journal of Intelligent Systems and Applications* (Vol. 8, no. 2, pp. 35–44). MECS Publisher. https://doi.org/10.5815/ijisa.2016.02.05

30. Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations (Version 3). *arXiv*. https://doi.org/10.48550/arXiv.2006.11477

31. Ma, J., & Schwartz, R. (2008). Unsupervised Versus Supervised Training of Acoustic Models. In *Interspeech* (pp. 2374–2377). https://doi.org/10.21437/interspeech.2008-122

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 (ASRU)*. IEEE. https://doi.org/10.1109/asru.2013.6707758

33. TensorFlow. (2020). The Functional API. https://www.tensorflow.org/guide/keras/functional

34. Apache Spark. (2020). ML Pipelines. https://spark.apache.org/docs/latest/ml-pipeline.html

35. Apache Airflow. (2020). DAGs. https://airflow.apache.org/docs/stable/concepts.html

36. Modin. (2020). Scale Your Pandas Workflow by Changing a Single Line of Code. https://modin.readthedocs.io/en/latest/

37. Lyudovyk, T., & Pylypenko, V. (2014). Code-Switching Speech Recognition for Closely Related Languages. In *Workshop on Spoken Language Technologies for Under-Resourced* (pp. 1–6).

38. Lyudovyk, T., & Pylypenko, V. (2016). Bilingual Speech Recognition without Preliminary Language Identification (pp. 12–34).

39. Vasileva, N., Pilipenko, V., Radutsky, A., Robeyko, V., & Sazhok, N. (2012). Corpus of Ukrainian on-Air Speech. In *Speech Technology* (Vol. 2, pp. 12–21).

40. Meyer, J. (2020). Open Speech Corpora. https://github.com/JRMeyer/open-speech-corpora

41. Xu, Q., Likhomanenko, T., Kahn, J., Hannun, A., Synnaeve, G., & Collobert, R. (2020). Iterative Pseudo-Labeling for Speech Recognition (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.2005.09267

42. Pa Pa Win, H., & Thu Thu Khine, P. (2020). Emotion Recognition System of Noisy Speech in Real World Environment. In *International Journal of Image, Graphics and Signal Processing* (Vol. 12, no. 2, pp. 1–8). MECS Publisher. https://doi.org/10.5815/ijigsp.2020.02.01

43. Kumar, J. A., Balakrishnan, M., & Wan Yahaya, W. A. J. (2016). Emotional Design in Multimedia Learning: How Emotional Intelligence Moderates Learning Outcomes. In *International Journal of Modern Education and Computer Science* (Vol. 8, no. 5, pp. 54–63). MECS Publisher. https://doi.org/10.5815/ijmecs.2016.05.07

44. Dhar, P., & Guha, S. (2021). A System to Predict Emotion from Bengali Speech. In *International Journal of Mathematical Sciences and Computing* (Vol. 7, no. 1, pp. 26–35). MECS Publisher. https://doi.org/10.5815/ijmsc.2021.01.04

45. Shirani, A., & Nilchi, A. R. N. (2016). Speech Emotion Recognition based on SVM as Both Feature Selector and Classifier. In *International Journal of Image, Graphics and Signal Processing* (Vol. 8, no. 4, pp. 39–45). MECS Publisher. https://doi.org/10.5815/ijigsp.2016.04.05

46. Devi, J. S., Yarramalle, S., & Prasad Nandyala, S. (2014). Speaker Emotion Recognition based on Speech Features and Classification Techniques. In *International Journal of Image, Graphics and Signal Processing* (Vol. 6, no. 7, pp. 61–77). MECS Publisher. https://doi.org/10.5815/ijigsp.2014.07.08

47. Abdel-Hamid, L. (2020). Egyptian Arabic Speech Emotion Recognition using Prosodic, Spectral and Wavelet Features. In *Speech Communication* (Vol. 122, pp. 19–30). Elsevier BV. https://doi.org/10.1016/j.specom.2020.04.005

48. Pajupuu, H. (2012). Estonian Emotional Speech Corpus. Center of Estonian Language Resources. https://doi.org/10.15155/EKI.000A

49. Kerkeni, L., Cleder, C., Serrestou, Y., & Raoof, K. (2020). French Emotional Speech Database – Oréau (Version 1) [Dataset]. *Zenodo*. https://doi.org/10.5281/zenodo.4405783

50. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B. (2005). A Database of German Emotional Speech. In *Interspeech 2005*. ISCA. https://doi.org/10.21437/interspeech.2005-446

51. Vrysas, N., Kotsakis, R., Liatsou, A., Dimoulas, C., & Kalliris, G. (2018). Speech Emotion Recognition for Performance Interaction. In *Journal of the Audio Engineering Society* (Vol. 66, no. 6, pp. 457–467). Audio Engineering Society. https://doi.org/10.17743/jaes.2018.0036

52. Vryzas, N., Matsiola, M., Kotsakis, R., Dimoulas, C., & Kalliris, G. (2018). Subjective Evaluation of a Speech Emotion Recognition Interaction Framework. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion* (Vol. 38, pp. 1–7). AM'18: Sound in Immersion and Emotion. ACM. https://doi.org/10.1145/3243274.3243294

53. Mohamad Nezami, O., Jamshid Lou, P., & Karami, M. (2018). ShEMO: A Large-Scale Validated Database for Persian Speech Emotion Detection. In *Language*

Resources and Evaluation (Vol. 53, no. 1, pp. 1–16). Springer Science and Business Media LLC. https://doi.org/10.1007/s10579-018-9427-x

54. Latif, S., Qayyum, A., Usman, M., & Qadir, J. (2018). Cross Lingual Speech Emotion Recognition: Urdu vs. Western Languages. In *2018 International Conference on Frontiers of Information Technology (FIT)*. IEEE. https://doi.org/10.1109/fit.2018.00023

55. Roberts, F., Margutti, P., & Takano, S. (2011). Judgments Concerning the Valence of Inter-Turn Silence Across Speakers of American English, Italian, and Japanese. In *Discourse Processes* (Vol. 48, no. 5, pp. 331–354). Informa UK Limited. https://doi.org/10.1080/0163853x.2011.558002

56. Neumann, M., & Thang Vu, N. goc. (2018). CRoss-Lingual and Multilingual Speech Emotion Recognition on English and French. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. https://doi.org/10.1109/icassp.2018.8462162

57. Li, X., & Akagi, M. (2019). Improving Multilingual Speech Emotion Recognition by Combining Acoustic Features in a Three-Layer Model. In *Speech Communication* (Vol. 110, pp. 1–12). Elsevier BV. https://doi.org/10.1016/j.specom.2019.04.004

58. Zehra, W., Javed, A. R., Jalil, Z., Khan, H. U., & Gadekallu, T. R. (2021). Cross Corpus Multi-Lingual Speech Emotion Recognition using Ensemble Learning. In *Complex & Intelligent Systems* (Vol. 7, no. 4, pp. 1845–1854). Springer Science and Business Media LLC. https://doi.org/10.1007/s40747-020-00250-4

59. Heracleous, P., & Yoneyama, A. (2019). A Comprehensive Study on Bilingual and Multilingual Speech Emotion Recognition using a Two-Pass Classification Scheme. In S. R. Shahamiri (Ed.), *PLOS ONE* (Vol. 14, no. 8, p. e0220386). Public Library of Science (PLoS). https://doi.org/10.1371/journal.pone.0220386

60. Sagha, H., Matějka, P., Gavryukova, M., Povolny, F., Marchi, E., & Schuller, B. (2016). Enhancing Multilingual Recognition of Emotion in Speech by Language Identification. In *Interspeech 2016*. ISCA. https://doi.org/10.21437/interspeech.2016-333

61. Scotti, V., Galati, F., Sbattella, L., & Tedesco, R. (2021). Combining Deep and Unsupervised Features for Multilingual Speech Emotion Recognition. In *Lecture Notes*

in Computer Science (pp. 114–128). Springer International Publishing. https://doi.org/10.1007/978-3-030-68790-8_10

62. Lech, M., Stolar, M., Best, C., & Bolia, R. (2020). Real-Time Speech Emotion Recognition Using a Pre-Trained Image Classification Network: Effects of Bandwidth Reduction and Companding. In *Frontiers in Computer Science* (Vol. 2). Frontiers Media SA. https://doi.org/10.3389/fcomp.2020.00014

63. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Prototyping Methodology of End-to-End Speech Analytics Software. In *4th International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS)* (Vol. 3312, pp. 76–86).

64. 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 (PIC S&T)* (pp. 335–337). https://doi.org/10.1109/picst51311.2020.9468084

65. Sun, M., Jiang, B., & Yuan, J. (2012). Vocal Emotion Recognition based on HMM and GMM for Mandarin Speech. In *International Journal of Education and Management Engineering* (Vol. 2, no. 3, pp. 25–31). MECS Publisher. https://doi.org/10.5815/ijeme.2012.03.04

66. Zia, T., Abbas, Q., & Akhtar, M. P. (2015). Evaluation of Feature Selection Approaches for Urdu Text Categorization. In *International Journal of Intelligent Systems and Applications* (Vol. 7, no. 6, pp. 33–40). MECS Publisher. https://doi.org/10.5815/ijisa.2015.06.03

67. Malik, P., & Singh Baghel, A. (2019). Performance Enhancement of Machine Translation Evaluation Systems for English-Hindi Language Pair. In *International Journal of Modern Education and Computer Science* (Vol. 11, no. 2, pp. 42–49). MECS Publisher. https://doi.org/10.5815/ijmecs.2019.02.06

68. Iosifov, I., Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Natural Language Technology to Ensure the Safety of Speech Information. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II)* (vol. 3187(1), pp. 216–226).

69. Iosifova, O., Iosifov, I., Rolik, O., & Sokolov, V. (2020). Techniques Comparison for Natural Language Processing. In *2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS)* (Vol. 2631(I), pp. 57–67).

70. Iosifova, O., Iosifov, I., & Rolik, O. (2020). Methods and Components of Natural Language Processing. In *Adaptive Automatic Control Systems* (Vol. 1, no. 36, pp. 93–113). Kyiv Politechnic Institute. https://doi.org/10.20535/1560-8956.36.2020.209780

71. Iosifova, O., Iosifov, I., Sokolov, V., Romanovskyi, O., & Sukaylo, I. (2021). Analysis of Automatic Speech Recognition Methods. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS)* (vol. 2923, pp. 252–257).

72. Shen, P., Lu, X., Li, S., & Kawai, H. (2020). Knowledge Distillation-Based Representation Learning for Short-Utterance Spoken Language Identification. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (Vol. 28, pp. 2674–2683). IEEE. https://doi.org/10.1109/taslp.2020.3023627

73. Gazeau, V., & Varol, C. (2018). Automatic Spoken Language Recognition with Neural Networks. In *International Journal of Information Technology and Computer Science* (Vol. 10, no. 8, pp. 11–17). MECS Publisher. https://doi.org/10.5815/ijitcs.2018.08.02

74. Almutiri, T., & Nadeem, F. (2022). Markov Models Applications in Natural Language Processing: A Survey. In *International Journal of Information Technology and Computer Science* (Vol. 14, no. 2, pp. 1–16). MECS Publisher. https://doi.org/10.5815/ijitcs.2022.02.01

75. Draghici, A., Abeßer, J., & Lukashevich, H. (2020). A Study on Spoken Language Identification using Deep Neural Networks. In *Proceedings of the 15th International Audio Mostly Conference* (pp. 253–256). AM'20: Audio Mostly 2020. ACM. https://doi.org/10.1145/3411109.3411123

76. Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech 2020*. ISCA. https://doi.org/10.21437/interspeech.2020-2650

77. Miao, X., McLoughlin, I., & Yan, Y. (2019). A New Time-Frequency Attention Mechanism for TDNN and CNN-LSTM-TDNN, with Application to Language Identification. In *Interspeech 2019*. ISCA. https://doi.org/10.21437/interspeech.2019-1256

78. Koluguri, N. R., Park, T., & Ginsburg, B. (2021). TitaNet: Neural Model for Speaker Representation with 1D Depth-Wise Separable Convolutions and Global Context (Version 1). *arXiv*. https://doi.org/10.48550/arXiv.2110.04410

79. Jia, F., Koluguri, N. R., Balam, J., & Ginsburg, B. (2022). A Compact End-to-End Model with Local and Global Context for Spoken Language Identification (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.2210.15781

80. Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., & Auli, M. (2021). XLS-R: Self-Supervised Cross-lingual Speech Representation Learning at Scale (Version 3). *arXiv*. https://doi.org/10.48550/arXiv.2111.09296

81. Conneau, A., Bapna, A., Zhang, Y., Ma, M., von Platen, P., Lozhkov, A., Cherry, C., Jia, Y., Rivera, C., Kale, M., Van Esch, D., Axelrod, V., Khanuja, S., Clark, J. H., Firat, O., Auli, M., Ruder, S., Riesa, J., & Johnson, M. (2022). XTREME-S: Evaluating Cross-lingual Speech Representations (Version 3). *arXiv*. https://doi.org/10.48550/arXiv.2203.10752

82. Nieto, O., Jin, Z., Dernoncourt, F., & Salamon, J. (2023). Efficient Spoken Language Recognition via Multilabel Classification. In *Interspeech 2023*. ISCA. https://doi.org/10.21437/interspeech.2023-1986

83. Sadjadi, S. O., Kheyrkhah, T., Tong, A., Greenberg, C., Reynolds, D., Singer, E., Mason, L., & Hernandez-Cordero, J. (2018). The *2017 NIST Language Recognition Evaluation. In The Speaker and Language Recognition Workshop (Odyssey 2018)* (pp. 82–89). ISCA. https://doi.org/10.21437/odyssey.2018-12

84. Valk, J., & Alumae, T. (2021). VoxLingua107: A Dataset for Spoken Language Recognition. In 2021 IEEE Spoken Language Technology Workshop (SLT). In *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. https://doi.org/10.1109/slt48900.2021.9383459

85. Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2019). Common Voice: A Massively-Multilingual Speech Corpus (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.1912.06670

86. Kukk, K., & Alumäe, T. (2022). Improving Language Identification of Accented Speech (Version 3). *arXiv*. https://doi.org/10.48550/arXiv.2203.16972

87. Bartley, T. M., Jia, F., Puvvada, K. C., Kriman, S., & Ginsburg, B. (2022). Accidental Learners: Spoken Language Identification in Multilingual Self-Supervised Models (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.2211.05103

РОЗДІЛ 3
МЕТОДИ СЕГМЕНТАЦІЇ, РОЗПІЗНАВАННЯ ТА ПІДВИЩЕННЯ ТОЧНОСТІ ОБРОБКИ ПРИРОДНОЇ МОВИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ПІДПРИЄМСТВА

3.1. Вимоги до даних для навчання мовних моделей

Підготовка даних є важливим кроком у будь-якій галузі NLP, і NLP не є винятком. Оскільки методи DL полягають у запам’ятовуванні та узагальненні навчального набору даних, важко переоцінити вплив якісного та неякісного набору даних. Як і в будь-яких інших завданнях ML, дані представлені у вигляді ознак і відповідних міток.

3.1.1. Вимоги до даних для обробки природної мови

Вимоги до вхідних даних такі ж, як і для інших завдань у галузі DL: вхідні дані мають бути максимально наближені до предметної області, в якій працюватиме модель. Простими словами, ви не можете навчити модель для прогнозування медичних анамнезів, використовуючи набір фінансових даних. Якщо в процесі навчання модель не побачила приклади токенів/слів у наборі даних, вона просто не відреагує на них. За останні роки був досягнутий величезний прогрес у подоланні таких обмежень, і техніка вбудовування з попередньо навченими токенами дуже допомагає. Проте, важко переоцінити, наскільки кращою буде навчена модель, якщо ви використовуєте релевантні навчальні дані.

Область NLP має деякі специфічні вимоги: дані повинні бути розділені, найчастіше на речення. Що є великою проблемою для ASR. Для таких специфічних завдань NLP, як пунктуація, загальним випадком є створення синтетичного набору даних і його маркування у найбільш відповідний для завдання спосіб [1].

Загалом, на сьогоднішній день існує велика кількість розмічених і набагато більше нерозмічених наборів даних, що є гарною відправною точкою для більшості завдань, тому інженерам-програмістам не потрібно збирати і маркувати набори даних самостійно.

3.1.2. Аналіз доступних мовних корпусів для української мови

Набори даних для ASR – це певна кількість аудіофайлів (зазвичай, 3–20) і пов'язані з ними текстові транскрипції (мітки). Для того, щоб модель звикла до акустики, всі цифри повинні бути де-нормалізовані до текстового представлення, щоб «4» і «format» звучали майже однаково, а якщо в навчальному наборі даних будуть «4» і «format», то моделі буде набагато важче узагальнювати акустично. Таке завдання де-нормалізації може бути дуже складним. Уявімо собі число «3», яке може означати «три», «третій» тощо. Для неанглійських мов таке завдання є ще складнішим.

Все вищесказане дає нам деяку інформацію про те, наскільки важко підготувати хороший набір даних для ASR, особливо для мов з низькими ресурсами. Існує кілька напрямків подолання таких обмежень, наприклад, (1) використовувати нерозмічені аудіодані (неконтрольоване навчання), що створює нові обмеження на обчислювальні ресурси і (2) генерувати такі набори даних ітеративно, використовуючи меншу кількість даних для створення більшої. Наприклад, в [2] ми згенерували 2 500 год. даних, використовуючи лише 100 год. як відправну точку. Як приклад мови для тестування такого автоматизованого конвеєра генерації наборів даних ASR ми взяли українську мову з низькими ресурсами, яка є дуже обмеженою з точки зору доступних наборів даних ASR (див. табл. 3.1).

Таблиця 3.1

Набори даних та джерела для українського мовного корпусу

Назва набору даних	Тип даних	Тривалість, год.	Якість
Український корпус мовлення для телерадіомовлення [3, 4]	Набір даних ASR	366	—
Багатомовний корпус «UkReco» [5]	Набір даних ASR	—	—
Український корпус M-AILABS [6]	Книги	87	Висока
Міністерство освіти, культури і науки [7]	YouTube	29	Середня
Deutsche Welle українською мовою [8]	YouTube	70	Достатня
Телебачення Торонто [9]	YouTube	60	Достатня
Спільний голос Mozilla [10]	Mozilla	22	Висока
TEDx Talks [11]	TEDx	<50	Середня

Однією з головних труднощів при підготовці даних для ASR є те, що багато аудіозаписів зберігаються в моноформаті, зі змішаними каналами, а отже, зі змішаними дикторами. Існує небагато методів відокремлення таких аудіозаписів за допомогою виявлення голосової активності та ідентифікації кількох дикторів.

Системи ASR досить вимогливі до даних. Для навчання хорошої моделі вам потрібен приблизно такий обсяг даних:

– 5 000 год. для гібридного підходу;
– 10 000 год. для комплексного підходу;
– 30 000 год. для навчання без нагляду [12].

3.2. Метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей

Підготовка даних є дуже важливим кроком у будь-якій задачі машинного та DL. У свою чергу, в процесі природної мови вхідні дані повинні бути сегментовані на речення. Сьогодні величезна кількість інформації зберігається в

аудіопослідовностях (окремо або як доповнення до відеопотоку). Багато з цих даних вже мають автоматично згенеровані транскрипти, які є наборами слів без розділових знаків та сегментації речень. Таким чином, це величезне сховище даних не може бути використане для вирішення завдань NLP у поточному стані. В даному розділі розглядаються методи підготовки сирих неформатованих транскриптів, щоб зробити ці дані доступними для подальшого використання в задачах NLP.

В [13] і [14] широко використовуються глибокі RNN та процес зворотного поширення. Трансформерна архітектура [15] була обрана як основна архітектура, як найпотужніша NLP для роботи з послідовностями короткої та середньої довжини, такими як текст. В [16–19] порівнювалися моделі та деякі інші.

Основний внесок поточної роботи в проблему сегментації неформатованого тексту полягає в пошуку ефективного підходу до використання транскрипції автоматизованого розпізнавання мови в суміжних областях NLP, таких як відповіді на запитання та автоматизація процесів.

3.2.1. Формулювання проблеми та вибір підходів до її вирішення

Як і для задач DL, для вирішення проблеми сегментації необробленого тексту існує багато можливих способів. Ми вирішили дослідити проблему за допомогою наступних підходів:

1. Задача мовного моделювання, а саме: за вхідною послідовністю (початком речення) спробуйте передбачити наступну лексему (слово або символ).

2. Задача маркування послідовності, яка полягає у тому, щоб присвоїти мітку кожній лексемі із заданої вхідної послідовності (у поточній задачі «поточна лексема – це остання лексема речення»).

Найпростіший спосіб розглянути проблему сегментації необробленого тексту – це завдання мовного моделювання, як показано на рис. 3.1, яке, передбачаючи наступну лексему, може також передбачати EOS.

Рис. 3.1. Демонстрація передбачення послідовності слів у пошуковій системі

Більшість робастних мовних моделей побудовано з використанням архітектури трансформера [15]. Навчання робастної мовної моделі є дуже ресурсномістким завданням, в той же час існує декілька можливих варіантів сучасних попередньо навчених моделей на основі архітектури трансформера, які можуть бути тонко налаштовані під конкретну задачу сегментації. Оскільки передбачення наступного токена не може бути вирішене за допомогою попередньо навчених реалізацій трансформера на основі енкодера (BERT, XLNet, ERNIE), а лише за допомогою повної архітектури трансформера або реалізації на основі декодера (Transformer-XL, GPT, GPT-2) [16–19], для поточної задачі було обрано попередньо навчену реалізацію на основі декодера.

Як показано на рис. 3.1, рухаючись слово за словом, архітектура трансформера на основі декодера (наприклад, GPT-2), яку також називають мовною моделлю, генерує наступний токен.

Основними змінами в доопрацюванні для поточної задачі було те, що модель повинна передбачати лише одну наступну лексему. Тому що немає сенсу передбачати більше одного, оскільки завдання полягає не в тому, щоб згенерувати послідовність, а в тому, щоб передбачити або окреме речення в заданому місці, або просунутися вперед. Звідси два доопрацьовані компоненти:

1. Замість виведення ідентифікатора наступної лексеми, модель повинна виводити всі softmax ймовірності без усікання.

2. Не потрібні алгоритми пошуку, оскільки потрібно передбачити лише одну наступну лексему.

Іншим можливим підходом є постановка задачі сегментації як задачі маркування, що є загальною задачею для NLP (маркування на основі токенів та на

основі груп маркування: розпізнавання іменованих об’єктів і синтаксичних шматків).

Для вирішення завдань маркування можна застосовувати різні методи, навіть класичні підходи ML, такі як дерева рішень. Що стосується поточної задачі сегментації речень, то попередня послідовність повинна мати вплив на вихідну мітку або кожну лексему. Тому найкраще підійдуть RNN [13, 14] та архітектури на основі трансформерів. Оскільки в попередньому підході використовувалися попередньо навчені моделі, поточний підхід також буде оцінюватися з використанням попередньо навчених моделей.

На відміну від підходу мовного моделювання, підхід маркування послідовності зосереджений на наданні мітки кожному токену вхідної послідовності, а вихідна послідовність дорівнює вхідній послідовності. Тому для поточної задачі було обрано реалізацію на основі попередньо навченого енкодера (BERT, XLNet, DistilBERT) [16–19].

Основна робота з підготовки такої архітектури та попередньо навчених моделей полягає в додаванні одного (щонайменше) додаткового шару поверх попередньо навченої моделі та підготовці даних для навчання нових моделей.

Основне завдання нового шару – знайти ваги, за допомогою яких буде мінімізовано втрати між базовими прогнозами моделі та наданими для навчання мітками. Зазвичай для такого налаштування використовується лише кілька (зазвичай 3–5) циклів навчання, що є цілком доступним навіть для великих наборів даних.

З боку підготовки даних не існує єдиного правильного підходу для даного завдання. Було обрано маркування «останнє слово в реченні». Всі інші слова в реченні були позначені як «не останнє слово в реченні», а фактичні роздільники були вилучені з навчальних даних, оскільки з цього моменту завдання моделі правильно передбачати на основі сирого тексту полягає в тому, чи є поточне слово останнім у реченні чи ні, як показано в табл. 3.2.

Таблиця 3.2

Демонстрація підходу до маркування послідовностей

sentence	with	the	end	of	sentence	separator	<EOS>	Next	sentence
0	0	0	0	0	0	1	0	0	0

Крім того, щоб моделі було легше бачити, що знаходиться праворуч і ліворуч від фактичних останніх слів, дані були розбиті на блоки однакової довжини (було обрано 16 токенів) [1].

3.2.2. Показники оцінювання та набори даних

Оскільки поточне завдання є завданням бінарної класифікації (ставити роздільник речення чи ні), то найбільш підходящими метриками для вимірювання точності будуть метрики бінарної класифікації (F1 Score, Precision, Recall) (див. розділ 2.1.1).

Крім того, для забезпечення точності, як і в будь-яких ресурсомістких завданнях, вимірювався час виконання завдання і час навчання моделі (для маркування послідовностей). Щоб уникнути неправильної інтерпретації та зв'язку з апаратним забезпеченням, буде використано відносне порівняння часу на виконання завдання.

Одномовні корпуси Вікіпедії enwiki-20181001-corpus [20], витягнуті з дампів Вікіпедії. Було використано перші 20 000 параграфів, що відповідає 1 899 353 навчальним токенам, 474 839 валідаційним токенам і 263 800 тестовим токенам.

Приклад набору даних з розміченим останнім словом у реченні:

```
Text: [… universal happiness is to our own even more important however
has been the …]
Labels: [… 0 0 0 0 0 1 0 0 0 0 0 0 0 …]
```

З обчислювальної точки зору, всі моделі були навчені та оцінені за допомогою графічного процесора Tesla P4.

3.2.3. Експериментальне порівняння підходів для моделювання

Загалом, як показано на рис. 3.2, для поточної задачі підхід розв'язання проблеми як задачі моделювання мови показує значно нижчі результати порівняно з підходом маркування послідовностей.

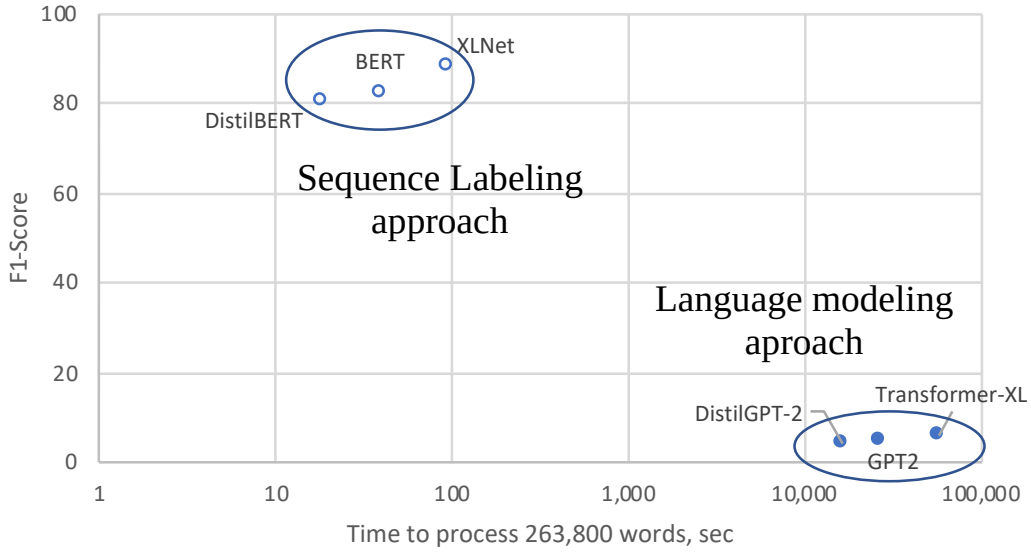


Рис. 3.2. F1-оцінки для груп моделей
при маркуванні послідовностей і мовному моделюванні

Крім того, суттєво поступається за обчислювальними витратами, як показано в табл. 3.3. Всі моделі, використані в підході маркування послідовностей, значно перевершили результати підходу мовного моделювання.

Таблиця 3.3

Порівняння розв'язання задачі сегментації речення
за допомогою різних підходів

Модель	Підхід	Трива-лість, сек	Прогно-зування часу, сек	F1-Score
DistilBERT	Маркування послідовності	1 309	18	80,40
База BERT	Маркування послідовності	3 121	39	82,10
XLNet	Маркування послідовності	3 504	92	88,35
DistilGPT-2	Мовне моделювання	–	16 270	4,02
GPT-2	Мовне моделювання	–	26 700	4,53
Трансформер-XL	Мовне моделювання	–	57 000	6,24

Якщо зосередитися на явних переможцях, підході маркування послідовностей і провести аналіз тут, то діапазон оцінки F1 склав майже 7,95%: від 80,40% для DistilBERT до 88,35% для XLNet. Це покращення точності оцінки F1 на 7,95% коштувало в п’ять разів більше обчислювального часу (18 секунд проти 92 секунд) між найшвидшим і найповільнішим, як показано на рис. 3.2.

Малі моделі, засновані на великих, такі як DistilBERT (на основі BERT), показують майже таку ж точність, при цьому займаючи на 58% менше часу на навчання і на 53% менше часу на прогнозування, ніж база BERT, що є значними показниками для виробничого середовища. Отже, якщо дельта в 2% точності оцінки F1 не є настільки значною, то рішення слід змістити на використання більш легкої моделі DistilBERT [21].

При мовному моделюванні підхід Transformer-XL не є суттєво кращим у прогнозуванні сепаратора, але потребує більше ніж у два рази більше часу, ніж GPT-2, як показано на рис. 3.2 і в табл. 3.3.

Важливо відзначити час навчання та час на підготовку набору даних для використання методу маркування послідовностей: для базової моделі BERT та моделі XLNet навчання зайняло 52 та 58 хв. відповідно. У той же час для DistilBERT це зайняло більш ніж у 2 рази менше часу – 21 хв. і 49 сек., як показано на рис. 3.3.

У той же час, мовне моделювання є набагато універсальнішим підходом і його легше реалізувати. Це означає, наприклад, що підхід з використанням послідовних міток може показувати погані результати на незвичних мітках, якщо вони не були належним чином навчені. Моделі підходу послідовних міток є дуже вузькими і специфічними для конкретних завдань/доменів, і тим, хто буде їх впроваджувати, слід пам’ятати про це.

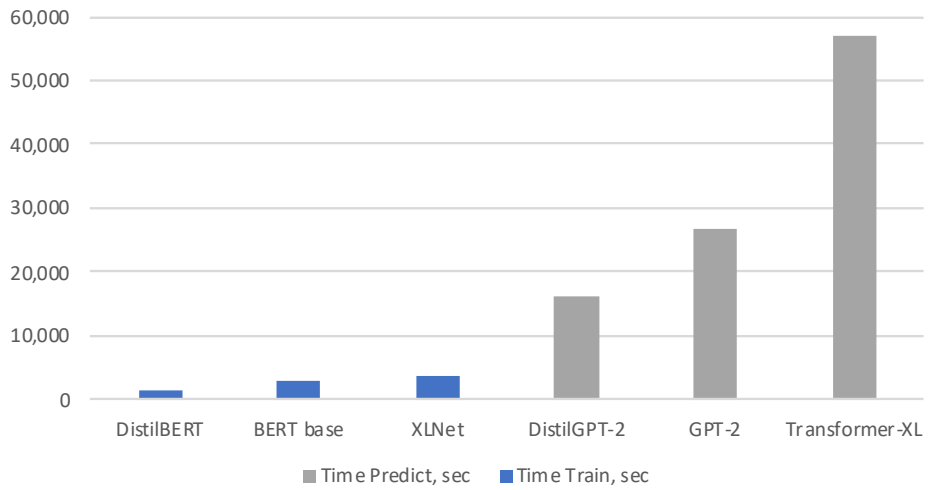


Рис. 3.3. Діаграма часу навчання TimeToTrain і прогнозування TimeToPredict (для 263 800 токенів)

Варто також зазначити, що підхід на основі мовної моделі показує кращі результати (до 72,4% F1-Score) на реальних даних Youtube (анотовані людиною транскрипції для відео на Youtube з 10 000 слів у якості тестового набору даних).

3.3. Метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами

Для розпізнавання багатомовних емоцій потрібно повести попередній підбір, підготовку та валідацію експериментальних наборів аудіоданих, на основі яких можна сформувати алгоритм і побудувати експериментальну установку та провести сам експеримент з верифікацією його результати.

3.3.1. Підбір наборів аудіоданих

Попередній список доступних наборів даних для розпізнавання мов (табл. 3.4) з різних груп (літерне позначення відповідно до ISO 639-1:2002 [22]):

– індоєвропейські: англійська (EN), німецька (DE), французька (FR), перська (FA) та урду (UR);

– уральська: естонська (ET);

– китайсько-тибетська: китайська (ZN).

Таблиця 3.4

Порівняння наборів даних для розпізнавання мовлення

Мова	Набір даних	Трива-лість, год.	Розмір	Модальності	Емоції
ZN	ESD [23]	29	7 000	Природня мова, голос	Нейтральний, гнів, смуток, щастя, здивування
DE	EMODB [24]	15	800	Природня мова, голос	Нейтральний, гнів, сум, щастя, страх, відраза, нудьга
ET	EKORPUS [25]	65	1 234	Природня мова, голос	Нейтральний, гнів, смуток, щастя
EN	CREMA [26]	203	7 442	Мультимодальні, голосові та візуальні	Нейтральний, гнів, сум, щастя, страх, відраза, огида
EN	IEMOCAP [27]	336	10 040	Мультимодальні, голосові та візуальні	Нейтральний, гнів, смуток, радість, страх, відраза, здивування, хвилювання, розчарування
EN	RAVDESS [28]	36	7 356	Мультимодальні, голосові та візуальні	Нейтральний, гнів, смуток, щастя, страх, відраза, здивування, спокій
EN	SAVEE [29]	19	480	Мультимодальні, голосові та візуальні	Нейтральний, гнів, смуток, радість, страх, відраза, здивування
EN	TESS [30]	55	2 800	Природна мова, голос	Нейтральний, гнів, смуток, радість, страх, відраза, здивування
FA	ShEMO [31]	196	3 000	Природна мова, голос	Нейтральний, гнів, смуток, щастя, страх, здивування
FR	OREAU [32]	23	482	Природна мова, голос	Нейтральний, гнів, смуток, радість, страх, відраза, здивування
UR	URDU [33]	16	400	Природна мова, голос	Нейтральний, гнів, смуток, щастя

З наведеного вище списку найбільше вражає різниця в розмірі. Набори даних повинні бути порівнянними за розміром для цілей навчання та оцінювання. Найменші набори даних складаються лише з 15–20 год. даних, і для того, щоб бути подібними, інші набори даних повинні бути скорочені до такої ж кількості даних.

Після оцінки були обрані наступні набори даних, які були скорочені до порівнянних розмірів для проведення експериментів (див. табл. 3.5).

Таблиця 3.5

Тривалість звуку в наборі даних

Мова	Набір даних	Тривалість аудіо на одну емоцію, год.			
		Нейтральний	Сердитий	Сумний	Щасливий
ZN	ESD	4	4	4	4
DE	EMODB	3	4	4	3
ET	EKORPUS	4	4	4	4
EN	SAVEE	4	3	4	3
FA	ShEMO	4	4	4	4
FR	OREAU	4	4	4	4
UR	URDU	4	4	4	4

Оскільки порівняння та оцінка типів архітектур на основі DNN не є метою даного дослідження, ми спиратимемося на дослідження в галузі вилучення інформації, пов’язаної з диктором. Ми використаємо нейронну мережу з акцентованою увагою, поширенням та агрегацією каналів у часових затримках (ECAPA-TDNN) [34].

Оцінки TDNN для задач SER можна знайти в роботі [35], яка показує, що архітектури TDNN дуже ефективні для прогнозування емоцій, а ECAPA-TDNN перевершує архітектуру TDNN на основі *x*-вектору.

3.3.2. Побудова експериментальної установки

Інструментарій SpeechBrain [36] було обрано як OpenSource з відповідною ліцензією та підтримкою обраної архітектури ECAPA-TDNN для прискорення

експериментів. Після декількох пробних оцінок, наступна архітектура виявилася найшвидшою та найточнішою. Архітектура моделі має наступний вигляд:

```
input_size: 96
channels: [512, 512, 512, 512, 1536]
kernel_sizes: [5, 3, 3, 3, 1]
dilations: [1, 2, 3, 4, 1]
attention_channels: 64
lin_neurons: 96
```

Найменша втрата була приблизно в 23-й епосі. Тому ми обмежуємо тренування 30 епохами:

```
number_of_epochs: 30
```

Залежно від експерименту кількість класів прогнозування варіюється від двох до чотирьох (гнів, нейтральність, щастя, смуток – починаючи з використання лише двох і закінчуючи чотирма):

```
out_n_neurons: 4
```

3.3.3. Підготовка та валідація експериментальних даних

Ми підготували дані в такий спосіб. По-перше, ми вирішили провести три експерименти для наборів даних з двома, трьома та чотирма емоціями, щоб оцінити, наскільки складніше для моделі розпізнавати три та чотири емоції, ніж просто бінарну класифікацію для нейтрального та гнівного стану. Ми вирішили використовувати наступні набори почуттів для кожного етапу експериментів:

– набір даних з двома емоціями складатиметься з нейтральних та гнівних емоцій;

– набір даних з трьома емоціями – з нейтральних, гнівних та сумних емоцій;

– набір даних про чотири емоції – з нейтральних, гнівних, сумних та щасливих почуттів.

Ми вирішили взяти ці емоції з двох причин:

1. Вони найбільш практично затребувані бізнесом.

2. Вони порівняно відрізняються один від одного акустично.

Для кожного з цих наборів експериментів ми створили окреме сховище (каталог), бо вирішили не змішувати і не оцінювати моделі, навчені класифікувати дві емоції, з наборами даних, що містять три, а для почуттів – навпаки.

3.3.4. Формування алгоритму експериментальної установки

Ми вибрали мови для наборів даних [ZN, DE, ET, EN, FA, FR, UR]. Для кожної мови ми створили окреме сховище (каталог) у сховищі експериментальних наборів. Ми усікли великі набори даних до порівнянного розміру, щоб збалансувати мовні дані за однакову кількість годин (див. рис. 3.4).

Для тестування та валідації ми вирішили використовувати 15% та 15% від кожного мовного набору даних, випадково вибраних між емоціями, тобто ми не брали 15% від кожної емоції, а дозволили випадковий вибір у прикладах для валідації.

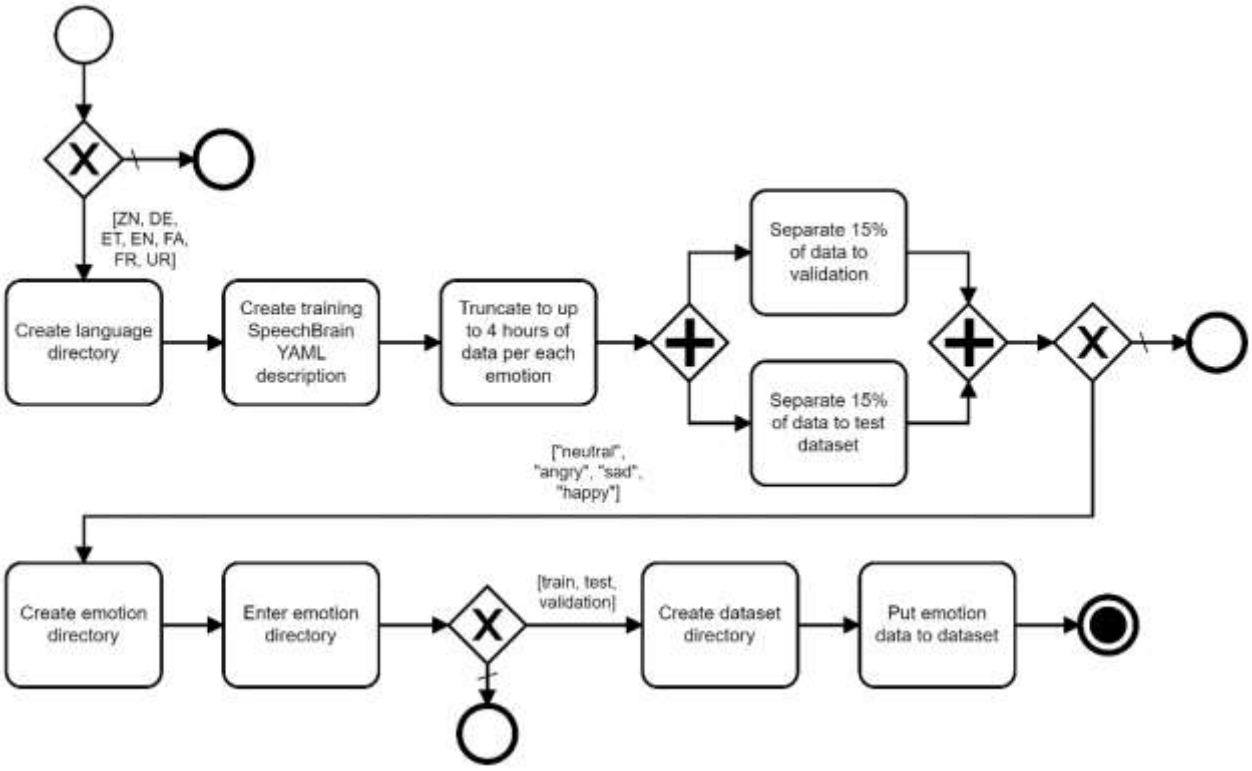


Рис. 3.4. BPMN-діаграма підготовки даних

Кожну емоцію ми помістили у сховище (каталог) з відповідним почуттям під відповідною мовою. На останньому кроці ми підготували опис SpeechBrain YAML для кожного набору експериментів, щоб мати правильну кількість вихідних нейронів.

3.3.5. Проведення експерименту за допомогою модельного тренінгу

Ми навчили 21 модель, по 7 моделей для кожного набору емоцій [нейтральний-злий], [нейтральний-злий-сумний] та [нейтральний-злий-сумний-щасливий].

Ми тренували моделі від двох до чотирьох емоцій (табл. 3.6) для кожної мови, використовуючи по чотири години аудіо для кожної емоції.

Таблиця 3.6

Експериментальні набори емоцій

Кількість емоцій	Набір емоцій	Модель, навчена на наборі даних
2	[нейтрально-злий]	ZN:ESD DE:EMODB ET:EKORPUS EN:SAVEE FA:ShEMO FR:OREAU UR:URDU
3	[нейтрально-злий-сумний]	
4	[нейтрально-злий-сумний-щасливий]	

Оцінювання проводилося для пари модель-мова шляхом ітерації від будь-якого до будь-якого. Таким чином, всі моделі були оцінені за всіма мовними наборами даних. Оцінювання проводилося з використанням валідаційного набору даних для кожної моделі, що складався з 15% для кожної емоції для кожної мови (до 36 хв.).

3.3.6. Верифікація результатів експерименту із розпізнання емоції

Було проведено 147 експериментів. По 49 експериментів для кожного набору емоцій [нейтральний-злий], [нейтральний-злий-сумний] та [нейтральний-злий-сумний-щасливий]. Кожен експеримент повторювався тричі для оцінки похибки відхилення. Стандартне відхилення оцінюється на рівні 2%.

У табл. 3.7–3.9 наведено результати оцінювання для наборів даних з двома, трьома та чотирма емоціями відповідно. У кожному стовпчику представлено модель мови-моделі, навчену на одному мовному наборі даних. У кожному рядку наведено результати оцінювання передбачення почуттів для набору мовних даних, вказаних у рядку, за допомогою моделі, натренованої у стовпчику.

На головній діагоналі ми бачимо результати оцінювання, коли модель і набір даних для оцінювання належать до однієї мови. Природно, що майже для всіх мов головна діагональ має найвище значення. Значний виняток становить естонська мова, яка показує не найкращі результати для естонського набору оціночних даних. Це може бути пов’язано з необробленою інтенсивністю емоцій, яка є відносно низькою для естонської мови. Ми обчислили середні оцінки і представили їх в останньому стовпчику та рядку.

Таблиця 3.7

Результати моделей оцінювання для різних мов для двох емоцій

Мова набору даних	Мова моделі							
	ZN	DE	ET	EN	FA	FR	UR	Медіана
ZN	0,94	0,73	0,73	0,73	0,73	0,73	0,73	0,73
DE	0,39	0,97	0,76	0,82	0,94	0,73	0,73	0,76
ET	0,46	0,46	0,62	0,46	0,50	0,54	0,62	0,50
EN	0,46	0,54	0,42	0,96	0,58	0,69	0,73	0,58
FA	0,38	0,52	0,57	0,38	0,90	0,62	0,71	0,57
FR	0,29	0,54	0,66	0,71	0,77	0,74	0,54	0,66
UR	0,41	0,64	0,82	0,41	0,79	0,62	0,98	0,64
Медіана	0,41	0,54	0,66	0,71	0,77	0,69	0,73	—

Таблиця 3.8

Результати моделей оцінювання різними мовами для трьох емоцій

Мова набору даних	Мова моделі							
	ZN	DE	ET	EN	FA	FR	UR	Медіана
ZN	0,96	0,60	0,53	0,53	0,58	0,51	0,64	0,58
DE	0,36	0,93	0,64	0,80	0,89	0,64	0,71	0,71
ET	0,27	0,30	0,46	0,30	0,46	0,43	0,27	0,30
EN	0,22	0,65	0,35	0,89	0,43	0,54	0,32	0,43
FA	0,23	0,57	0,53	0,43	0,77	0,43	0,43	0,43
FR	0,35	0,52	0,40	0,52	0,48	0,62	0,40	0,48
UR	0,22	0,33	0,52	0,28	0,34	0,36	0,97	0,34
Медіана	0,27	0,57	0,52	0,52	0,48	0,51	0,43	

Таблиця 3.9

Результати моделей оцінювання для різних мов для чотирьох емоцій

Мова набору даних	Мова моделі							
	ZN	DE	ET	EN	FA	FR	UR	Медіана
ZN	0,90	0,47	0,34	0,31	0,48	0,44	0,52	0,47
DE	0,39	0,73	0,49	0,41	0,53	0,36	0,25	0,41
ET	0,26	0,30	0,34	0,30	0,26	0,30	0,26	0,30
EN	0,27	0,45	0,33	0,86	0,33	0,37	0,39	0,37
FA	0,21	0,19	0,48	0,24	0,55	0,29	0,33	0,29
FR	0,28	0,27	0,30	0,23	0,25	0,66	0,23	0,27
UR	0,27	0,17	0,32	0,29	0,36	0,25	0,83	0,29
Медіана	0,27	0,30	0,34	0,30	0,36	0,36	0,33	

З таблиць можна простежити, наскільки важче передбачити три емоції, ніж дві, з медіаною точності на 18% нижчою в усіх мовах. Ми також бачимо, наскільки важче передбачити чотири емоції, ніж дві, з медіаною точності на 33% нижчою в усіх мовах.

На рис. 3.5–3.7 ми візуалізуємо медіану та стандартне відхилення для кожної навченої моделі, але виключаємо оцінювання тією ж мовою, якою було введено модель. Це означає, що ми виключили оцінювання моделі DE за допомогою набору даних оцінювання DE, щоб мати чітке уявлення про те, як працює кожна модель для ненавчених мов, що не переносяться моделями.

Як бачимо, китайська мова не перекладається на жодну мову, навіть для такої простої установки, як дві емоції. Ми також несподівано бачимо, наскільки стабільною у перекладі є модель, навчена фарсі.

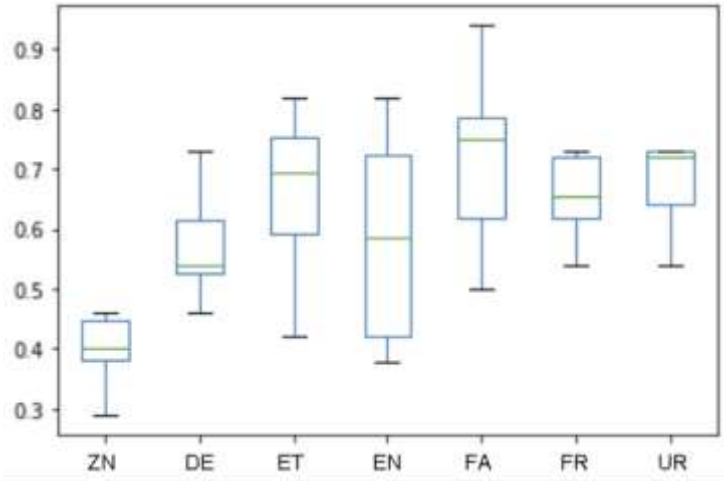


Рис. 3.5. Діаграма розмаху модельної мови для двох емоцій

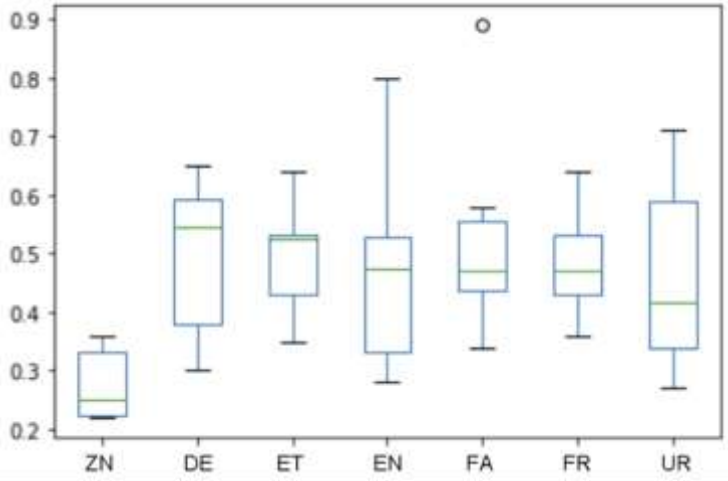


Рис. 3.6. Діаграма розмаху модельної мови для трьох емоцій

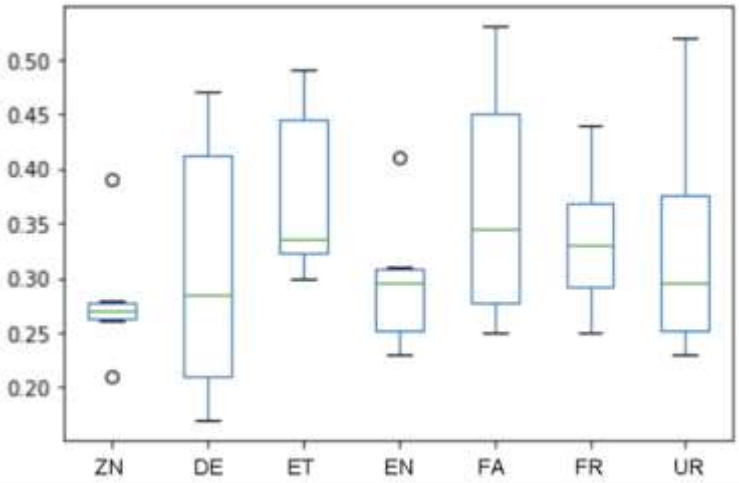


Рис. 3.7. Діаграма розмаху модельної мови для чотирьох емоцій

З табл. 3.7–3.9 також можна відстежити несподівану дзеркальну поведінку (див. табл. 3.10). Продемонструємо це на наступних двох парах для двох емоцій DE-FA та FR-ZN.

Таблиця 3.10

Тривалість аудіо з набору даних

Мова моделі	Мова оцінювання	Точність
Пара FR-ZN		
FR	ZN	0,73
ZN	FR	0,29
Пара DE-FA		
DE	FA	0,52
FA	DE	0,94

Це приводить нас до цікавих висновків: якщо емоції досить добре передаються з мови оригіналу на мову перекладу, це не означає, що емоції можуть так само добре передаватися у зворотному напрямку між мовами [37].

3.4. Метод підвищення точності розпізнавання природної мови для близькоспоріднених мов

3.4.1. Архітектура експериментальної установки

Ми використали SLI-систему, побудовану на глибокій архітектурі CNN NVIDIA TitaNet [38, 39], яка є частиною інструментарію Nemo [40], зображеного на рис. 3.8. Енкодер TitaNet-LID-BxRxC базується на архітектурі ContextNet, де B – кількість блоків, R – кількість повторюваних «базових» блоків, а фільтри в шарах згортки кожного блоку.

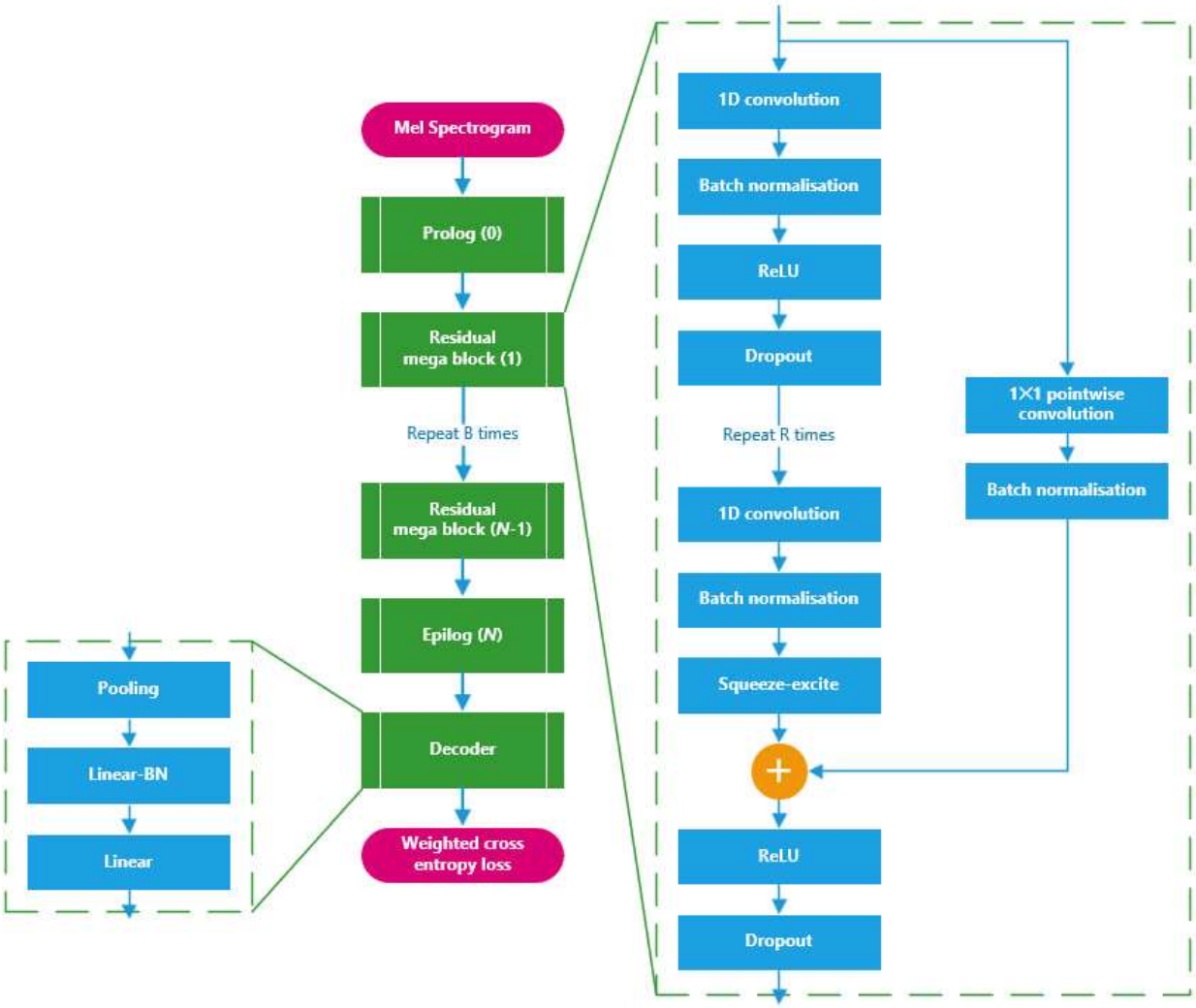


Рис. 3.8. Архітектура моделі ідентифікації природня мови

TitaNet-LID-BxRxC, як показано на рис. 3.8, має енкодер з одномірним розділенням каналів за глибиною, що відображає архітектуру ContextNet, і декодер. Цей енкодер складається з B блоків, де кожен блок містить три «базові» блоків, що повторюються, і C згорткових фільтрів у кожному блоці. Енкодер TitaNet-LID починається з початкового блоку B_0, а потім продовжується серією залишкових мегаблоків від B_1 до $B_N - 1$.

Кожен мегаблок складається з R «базових» блоків, які завершуються модулем стиснення-збудження. Базовий блок включає 1D-згортковий модуль, що відокремлюється за часовим каналом (на рис. 3.8 називається 1D-згорткою), з ядром K. За ним послідовно йдуть компоненти BatchNorm, ReLU та Dropout. Кожен модуль 1D-згортки складається з двох компонентів: шару згортки за глибиною та

шару згортки за точками 1×1. Ці базові блоки повторюються R разів і з'єднуються між собою за допомогою шарів «стиснення-збудження» (Squeeze-and-Excitation, SE), які включають глобальне усереднене об'єднання.

Ядро K у повторюваних «базових» блоках має номери 7, 11 і 15. Енкодер завершується фінальним блоком B_N, який виробляє проміжні звукові характеристики. Ці закодовані звукові ознаки потім передаються до декодера для класифікації. Декодер складається з двох частин: шару об'єднання статистики (обчислення середнього значення і стандартного відхилення), який перетворює вхідні звукові ознаки змінної довжини в представлення ознак фіксованої довжини, і двох лінійних шарів. Перший лінійний шар має вихідний розмір 512, тоді як другий лінійний шар виконує перетворення від 512 до кінцевої кількості класів N.

Всі моделі навчалися протягом 40 епох на одному вузлі з двома GPU з розміром партії 32 на GPU і точністю fp16. Ми використовували оптимізатор Adam та планувальник швидкості навчання Cosine Annealing з коефіцієнтом прогріву 10%. Максимальна швидкість навчання становила 0,001, а мінімальна – 0,0001. Для всіх фрагментів навчання було використано випадкове число 42. Всі експериментальні моделі навчалися за однаковою процедурою навчання.

3.4.2. Відбір та порівняння наборів даних

Найбільш широко визнані стандарти для оцінки нових моделей і технологій SLI/SLR базуються на наборах даних NIST LRE, але їхнім недоліком є те, що вони є досить дорогими. Саме тому в дослідженні ми встановили два критерії для наборів даних:

1. Вибрані набори даних мають бути відкритими.

2. Відібрані набори даних повинні охоплювати стільки мов, скільки їх існує у світі.

Після оцінки наявних наборів даних було обрано наступні набори даних: для навчання загальної моделі SLI ми використали великий багатомовний набір даних VoxLingua107 [41], а для покращення точності моделі – CV [42].

Навчальний набір даних VoxLingua107 складається з 6 628 год. мовлення, які були отримані з відео на YouTube шляхом автоматичного вилучення, як зазначено в [41]. Мова, визначена в назві та описі відео, класифікує ці мовленнєві сегменти за різними мовами. Для підвищення точності мовного маркування було застосовано метод пост-фільтрації на основі даних. Цей підхід допоміг усунути сегменти, які, ймовірно, не належать до певної мови. В результаті точність правильно розмічених сегментів у наборі даних зросла до 98%, що підтверджено оцінкою на основі краудсорсингу. VoxLingua107 містить записи мовлення 107 різними мовами, з приблизно 2,54 млн окремих мовних сегментів. Середня тривалість цих сегментів становить 9,4 сек., і кожна мова надає близько 62 год. даних. Підмножина розвитку VoxLingua107 включає 4,5 год. спонтанного мовлення, витягнутого з відео на YouTube, що охоплює 33 мови. У цій підмножині лінгвістична класифікація кожного мовного сегмента була підтверджена щонайменше двома носіями мови або висококваліфікованими дикторами, обраними з натовпу. Цей набір для розробки містить загалом 1 608 мовленнєвих сегментів.

Тестовий/оціночний набір даних VoxLingua107. Як набір для розробки з вручну перевіреними мовними мітками, він обмежений для VoxLingua107 і містить лише мітки для 33 мов. Цього було недостатньо для експериментів, тому ми вирішили розділити 10% навчального набору даних для цілей тестування/оцінювання.

Набір CV даних [42] – це обширна багатомовна збірка транскрибованих вокальних записів, призначена для досліджень і розробок у галузі мовних технологій. Цей набір даних охоплює 9 283 год. аудіозаписів. Крім того, він містить демографічні метадані, що містять таку інформацію, як вік, стать і акцент. Із загальної колекції 7 335 год. на 60 мовах були затверджені для використання.

Набір CV-Clean даних – це підмножина набору даних CV з верифікованими мітками. Верифікацію було виконано за допомогою автоматизованого пайплайну [2] у два етапи:

1. Розпізнавання мовлення за допомогою моделі Whisper v2 з відкритим вихідним кодом.

2. Фільтрація всіх сегментів, де мітка CV не дорівнює передбаченій міткою розпізнавача мовлення.

CV-Balanced набір даних – це підмножина CV/CV-Clean наборів даних, урізаних/збалансованих найменшим набором даних для пари/трійки експериментальних мов для певного регіону. Наприклад, якщо мова *A* має 500 год., а мова *B* – лише 100 год. аудіо, збалансований набір даних складатиметься лише з 100 год. для обох мов.

3.4.3. Вибір мов низької точності для проведення експериментів

Ми провели серію експериментів з наступними кроками, щоб отримати мови низької точності:

1. Навчіть загальну модель SLI для 107 мов, використовуючи набір даних VoxLingua107.

2. Визначити мови з низькою точністю.

3. Вибрати мовні пари для покращення якості SLI. На основі матриці заплутаності між мовами та наявності наборів даних CV.

4. Для кожної обраної пари:

– тренувати SLI, використовуючи підрозділ набору даних VoxLingua107 для вибраних пар, щоб підвищити точність обох мов у парах;

– тренувати SLI, використовуючи підрозділ наборів даних {VoxLingua107 + CV} для вибраних пар, щоб підвищити точність обох мов у парах;

– чистий набір даних резюме;

– тренувати SLI, використовуючи підрозділ наборів даних {VoxLingua107 + CV-Clean} для вибраних пар, щоб підвищити точність обох мов у парах.

Вибір мови та регіону для експериментів:

Крок 1. Відбір мов для експериментів за низькою точністю моделі, навченої на 107 мовах з набору даних VoxLingua107. Ми навчили базову модель для всіх 107 мов, представлених у VoxLingua107, щоб визначити, які мови не можуть бути навчені з високою точністю. Як згадувалося раніше, ми розділили навчальний набір даних і зарезервували 10% набору даних для оцінювання. Оцінка F1 для всіх 107 мов представлена на рис. 3.9. Для подальших експериментів нас цікавили лише мови з точністю менше 95%. Сорок мов були відібрані за порогом 95%.

Крок 2. Відбираємо мови для експериментів на основі доступності набору даних CV. На цьому кроці ми звужуємо список мов для експериментів, оцінюючи доступні набори даних CV для мов з Кроку 1. Нас цікавили пари з відкритими наборами даних, на яких можна було б навчатися. Тому ми оцінили наявність наборів даних CV для цих 40 мов, представлених у табл. 3.11. Для деяких мов набори даних відсутні.

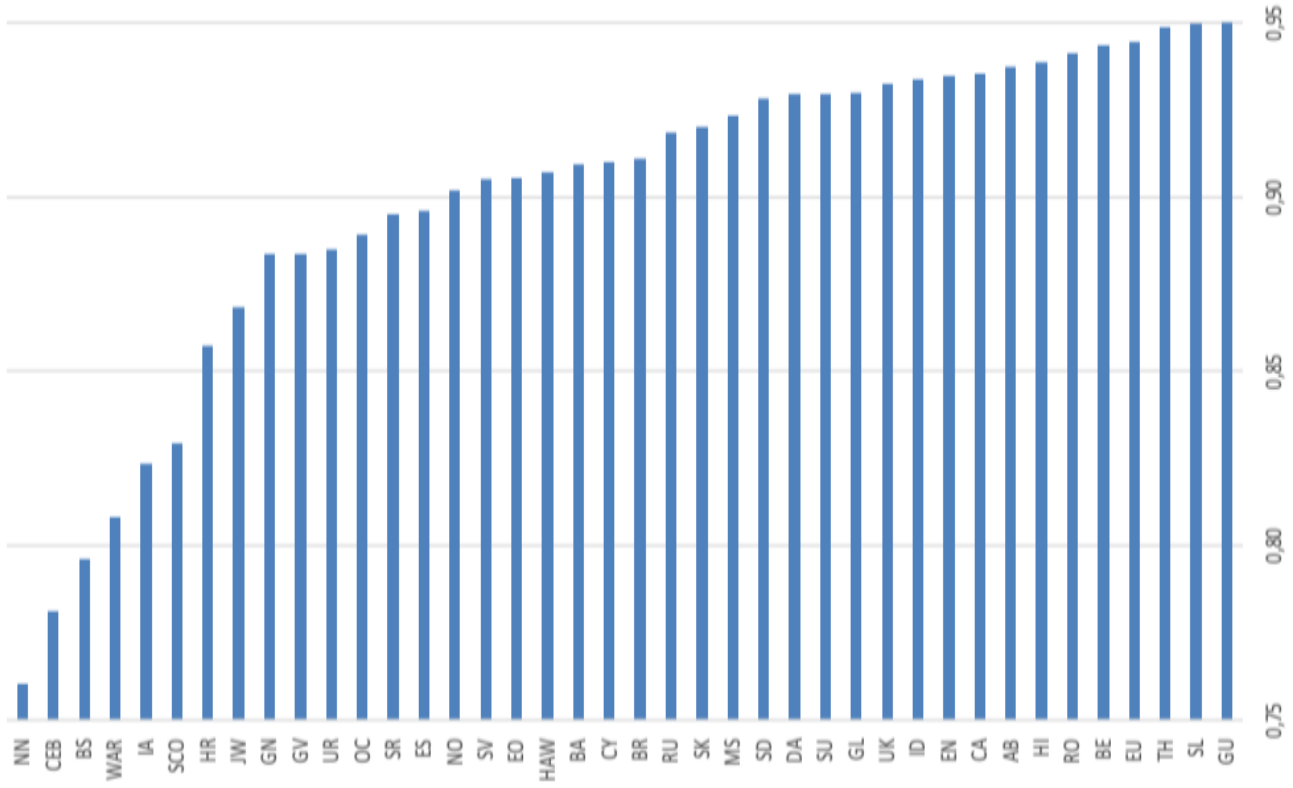


Рис. 3.9. Стовпчикова діаграма об’єму аудіоданих для різних мов

Таблиця 3.11

Порівняння оцінок F1 для мов низької точності моделі, навченої на VoxLingua107

Код	Мова	Оцінка F1, %	Код	Мова	Оцінка F1, %
GU	гуджаратська	95,0	CY	валлійська	91,0
SL	словенська	95,0	BA	башкирська	90,9
TH	тайська	94,9	HAW	гавайська	90,7
EU	баскська	94,4	EO	есперанто	90,5
BE	білоруська	94,3	SV	шведська	90,5
RO	румунська	94,1	NO	норвезька	90,2
HI	хінді	93,9	ES	іспанська	89,6
AB	абхазька	93,7	SR	сербська	89,5
CA	каталонська	93,5	OC	окситанська	88,9
EN	англійська	93,5	UR	урду	88,5
ID	індонезійська	93,4	GV	менська	88,4
UK	українська	93,3	GN	гуарані	88,4
GL	галісійська	93,0	JW	яванська	86,8
SU	сунданська	92,9	HR	хорватська	85,7
DA	данська	92,9	SCO	шотландська	82,9
SD	синдхі	92,8	IA	інтерлінгва	82,4
MS	малайзійська	92,3	WAR	варайська	80,8
SK	словацька	92,0	BS	боснійська	79,6
RU	російська	91,8	CEB	себуанська	78,1
BR	бретонська	91,1	NN	новонорвезька	76,0

Крок 3. Вибір мов для експериментів за матрицею заплутаності. Ми оцінили матрицю заплутаності для мов з наявними наборами даних у табл. 3.12. Матриця заплутаності – це таблиця, яка використовується для оцінки ефективності моделей класифікації. Вона організована таким чином, щоб показати порівняння між фактичною та передбачуваною класифікаціями. Матриця надає візуальний і числовий спосіб зрозуміти точність моделі, вказуючи, скільки прогнозів були правильними або неправильними, а також природу помилок. Ми будуємо матрицю помилок за допомогою бібліотеки Python sklearn.

Таблиця 3.12

Розмір навчальних наборів даних для окремих мов

Код	Зареєстровано, год.	Перевірено, год.	Кількість голосів
SL	14	11	146
TH	420	171	8
EU	159	105	1
BE	1 632	1 586	8 205
RO	44	19	408
HI	20	14	396
AB	85	60	400
CA	3 328	2 554	35 062
EN	3 347	2 532	88 904
ID	64	29	516
UK	105	94	1
GL	61	38	1
DA	13	12	243
MS	10	3	128
SK	27	22	216
RU	260	228	3
BR	26	12	196
CY	155	122	1 800
BA	268	258	912
EO	1 897	1 432	1 682
SV	54	45	808
ES	2 188	526	25 338
SR	6	4	144
OC	13	2	141
UR	221	63	316
GN	25	4	140
IA	17	14	66
NN	2	2	32

Нас цікавили значущі числа в недіагональних позиціях на матриці заплутаності. Ці точки показували, коли прогноз моделі був неправильним і яку мову вона передбачала замість правильної. Це дає змогу зрозуміти, як модель плутає мови, які пари/трійки найчастіше плутають між собою. Критерієм, за яким ми відбирали мови для експериментів, було те, що модель повинна була зробити більше 20 помилок для неправильної мови (див. рис. 3.10).

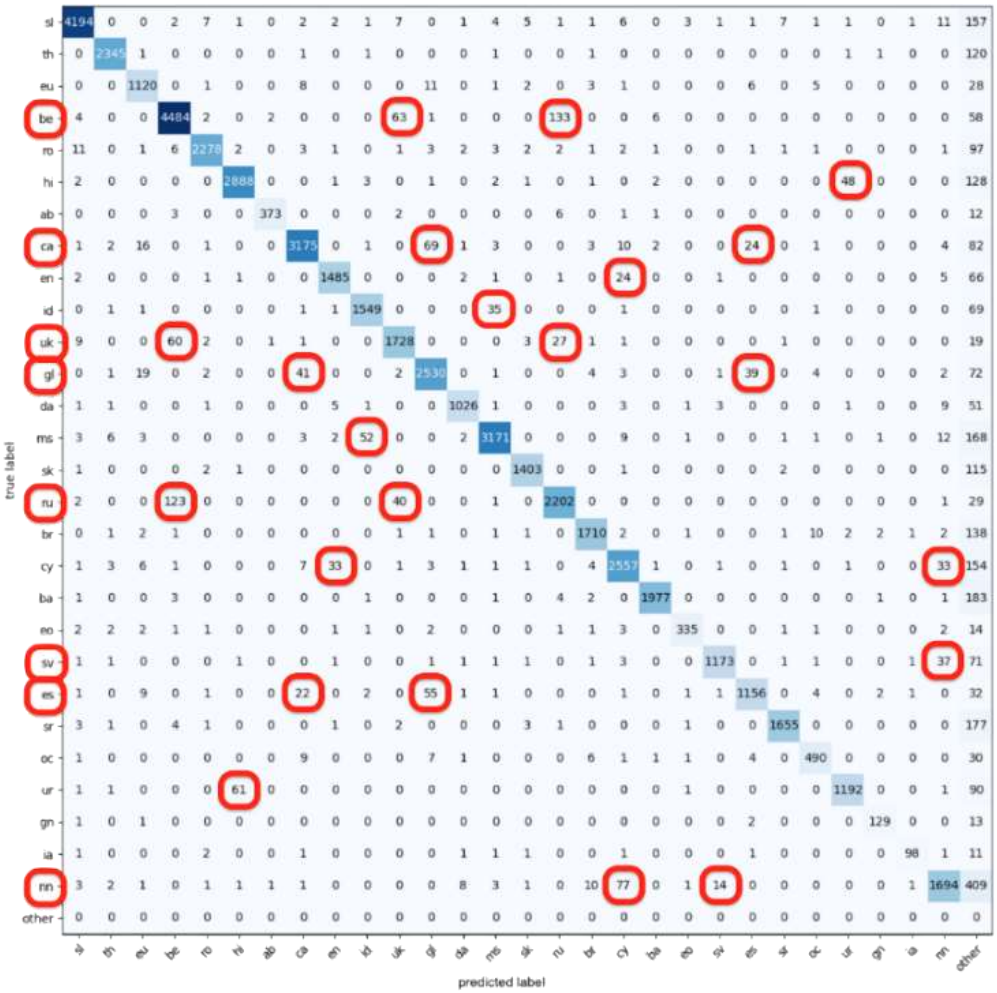


Рис. 3.10. Матриця заплутаності для мов низької точності

Оцінивши низьку точність і високу заплутаність, ми розробили пари/трійки «мова-регіон» для експерименту, представлені в табл. 3.13.

Таблиця 3.13

Вибір мови для експериментів

Мовний регіон	Мовний код	Оцінка VoxLingua107 F1, %	Плутають з, %
іспанська / каталонська / галісійська	ES → GL	89,6	4,3
	CA → GL	93,5	2,0
	GL → CA	93,0	1,5
шведська / новонорвезька	SV → NN	90,5	2,9
	NN → SV	76,0	0,6
українська / російська	UK → RU	93,3	1,5
	RU → UK	91,8	1,7
словацька / чеська	SK → CS	92,0	4,7
	CS → SK	97,1	1,4

Стовпчик «Плутають з» відображає відносну кількість помилок, коли модель помилково використовує неправильну мову. Наприклад, для іспанських аудіозразків модель у 4,3% випадків неправильно передбачила галісійську мову замість очікуваної іспанської.

3.4.4. Тренінг за допомогою набору тестових даних

Навчальні набори даних для експериментів представлено в табл. 3.14 з відповідними розмірами в хвилинах, годинах та кількістю вибірок, щоб зрозуміти дисбаланс наявних даних для мов регіону.

Таблиця 3.14

Розмір навчальних наборів даних для окремих мов

Мова	Розмір		Набір даних
	год.	екз.	
CS	60,0	23 215	Vox107
	25,5	19 358	CV
	12,3	9 656	CV-Cl
	3,5	2 729	CV-Bal
	1,4	1 128	CV-Cl-Bal
SK	35,8	13 727	Vox107
	3,5	3 276	CV
	1,4	1 389	CV-Cl
	3,5	3 276	CV-Bal
	1,4	1 389	CV-Cl-Bal
SV	30,6	11 475	Vox107
	8,4	7 584	CV
	4,9	4 638	CV-Cl
	0,5	496	CV-Bal
	—	1	CV-Cl-Bal
NO	96,0	37 408	Vox107
	0,5	407	CV
	—	1	CV-Cl
	0,5	407	CV-Bal
	—	1	CV-Cl-Bal
UK	47,2	16 976	Vox107
	22,5	19 024	CV
	10,6	9 313	CV-Cl

Мова	Розмір		Набір даних
	год.	екз.	
UK	22,5	19 024	CV-Bal
	10,6	9 313	CV-Cl-Bal
RU	65,6	21 412	Vox107
	37,7	26 328	CV
	24,8	17 789	CV-Cl
	22,5	15 854	CV-Bal
	10,6	7 487	CV-Cl-Bal
ES	34,7	11 475	Vox107
	450,8	311 392	CV
	297,3	206 270	CV-Cl
	16,5	10 057	CV-Bal
	4,9	2 987	CV-Cl-Bal
CA	79,2	30 460	Vox107
	1743,8	1 142 607	CV
	821,0	546 025	CV-Cl
	16,5	9 625	CV-Bal
	4,9	2 942	CV-Cl-Bal
GL	65,0	24 958	Vox107
	16,5	12 688	CV
	4,9	3 945	CV-Cl
	16,5	12 688	CV-Bal
	4,9	3 945	CV-Cl-Bal

В експериментах було використано п’ять наборів даних: VoxLingua107 (Vox107), CV, CV-Clean (CV-Cl), CV-Balanced (CV-Bal) та CV-Clean-Balanced (CV-Cl-Bal). Як видно з таблиці, існують певні обмеження щодо навчальних та тестових наборів даних. CV та CV-Clean для норвезької мови і, як наслідок, CV-Clean-Balanced для норвезької та шведської є мінімальними (менше однієї хвилини), що ми використовуємо в експерименті лише для узгодженості результатів графіків, але не є надійним для такої малої кількості даних.

3.4.5. Верифікація результатів експерименту із розпізнання двійок та трійок близькоспоріднених мов

По-перше, ми вирішили оцінити вплив на точність багатомовної моделі (модель на 107 мовах) порівняно з регіональними моделями (модель для кожного регіону на 2–3 мови). Це важливо, оскільки (1) часто компаніям не потрібна ціла модель на 107 мовах, а потрібна найточніша модель для конкретного регіону, і (2) найбільше заплутаності було виявлено в регіональних мовах (наприклад, іспанська, каталонська, галісійська), і це є основним фокусом даної роботи.

Обидві моделі були навчені та протестовані з використанням набору даних VoxLingua107:

– model-voxlingua-107 на 90% складається з VoxLingua107 для навчання і на 10% для тестування;

– model-voxlingua-regional (модель для кожного регіону для 2–3 мов) становить 90% від VoxLingua107 для регіональних мов (див. табл. 3.14) для навчання і 10% для тестування.

Як показано на рис. 3.11, моделі, явно навчені для 2–3 мов, демонструють значно вищу точність, ніж багатомовні (модель для 107 мов). Середнє покращення результату F1 становить 2,47%, а медіана – 1,65% в абсолютних числах.

Model-voxlingua-107 – це уніфікована модель SLI, навчена для 107 мов, а model-voxlingua-region – це регіональна модель SLI, навчена для конкретних 2–3 мов обраного регіону.

Щоб зрозуміти, наскільки точно модель працює на різних даних, ми оцінили точність моделей, навчених за регіонами, на наборах даних VoxLingua107, використовуючи набори даних CV.

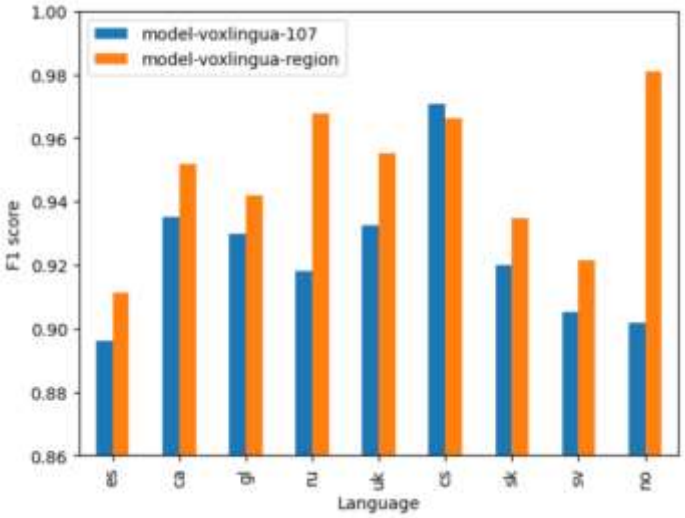


Рис. 3.11. Стовпчикова діаграма точності для моделі, навченої регіонами, порівняно з уніфікованою VoxLingua107 моделлю

Ми оцінювали лише моделі, навчені за регіонами, оскільки вони є найбільш точними. Як ми бачимо з діаграми точності на рис. 3.12, всі мови працюють набагато гірше на даних набору CV. Подивимося, чи можна покращити точність моделей.

Ми оцінювали моделі, навчені на оригінальному VoxLingua107, на тестовому наборі даних VoxLingua107 і для контрасту на зовнішньому наборі даних CV. Погіршення точності при оцінюванні на невидимому наборі даних CV становить в середньому 19,7%, а медіана – 17,6%.

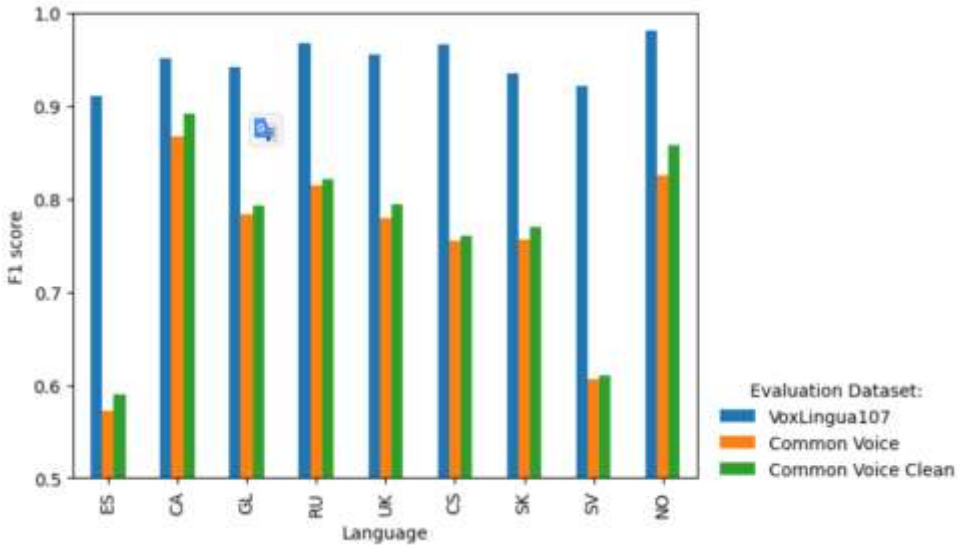


Рис. 3.12. Стовпчикова діаграма середньої деградація точності

Ми навчили п’ять моделей з комбінацією наборів даних: {Vox107}, {Vox107 + CV}, {Vox107 + CV-Clean}, {Vox107 + CV-Balanced} та {Vox107 + CV-Clean-Balanced}. Кожну з цих моделей ми оцінювали на тестових/оціночних наборах даних.

На рис. 3.13–3.15 ми бачимо оцінку точності F1 іспансько-каталонсько-галісійської мовної трійки.

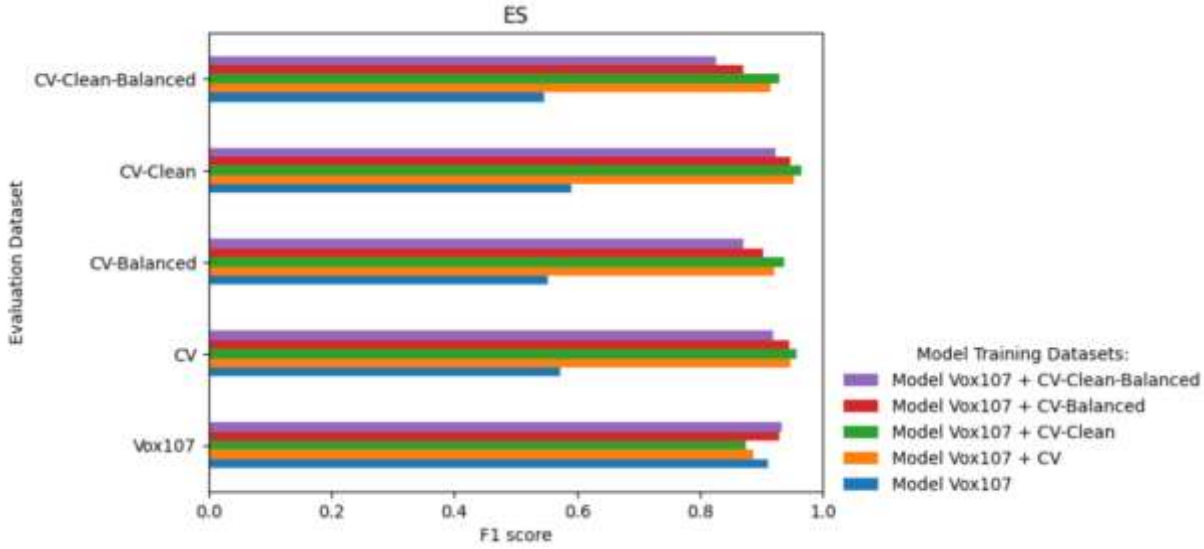


Рис. 3.13. Стовпчикова діаграма оцінки моделей для іспанської мови

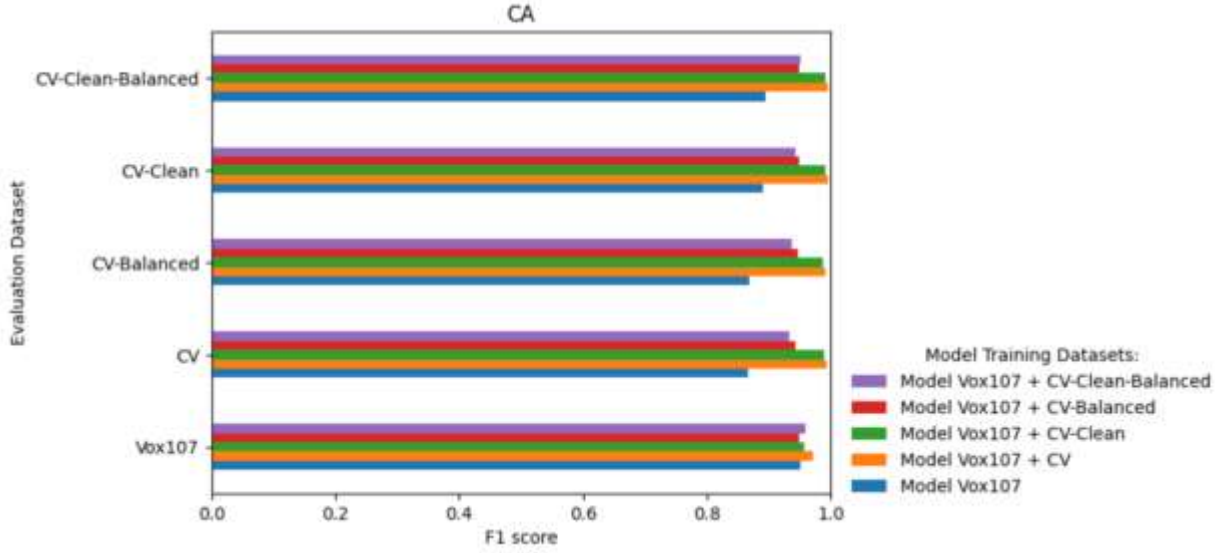


Рис. 3.14. Стовпчикова діаграма оцінки моделей для каталонської мови

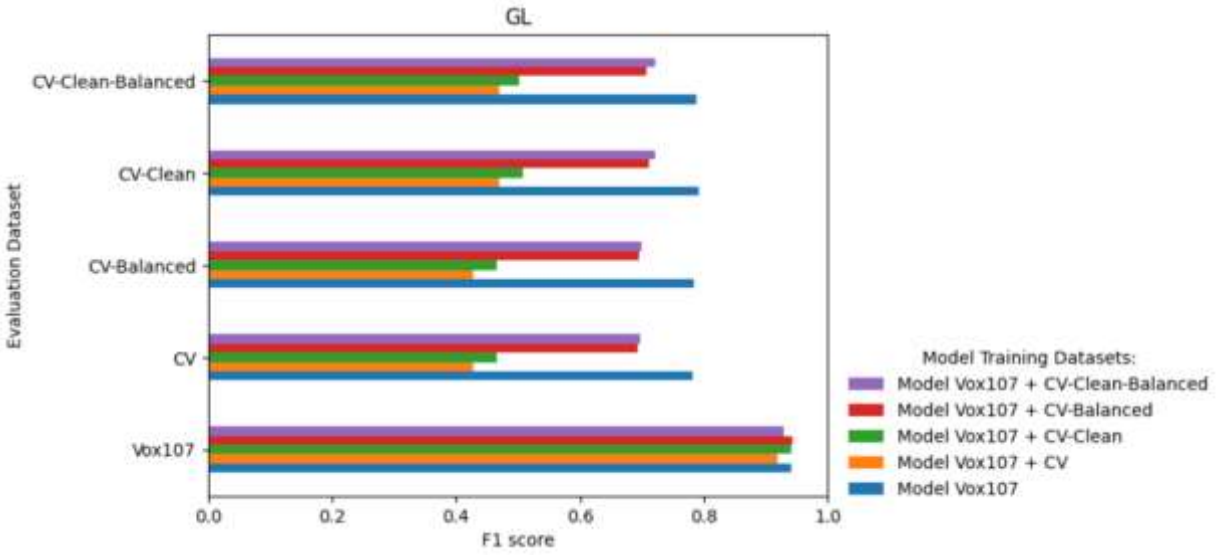


Рис. 3.15. Стовпчикова діаграма оцінки моделей для галісійської мови

Як видно з табл. 3.14, дані для галісійської мови значно незбалансовані порівняно з іспанською та каталонською мовами (CV 16,5 год. для галісійської мови проти 450 та 1743 для іспанської та каталонської мов відповідно). Це призводить до значного погіршення точності для галісійської мови в моделях, навчених на наборах даних з дисбалансом.

Збалансувавши набори даних за найменшим з них, ми досягли стабільних результатів на всіх трьох мовах – див. модель, навчену на {VoxLingua107 + CV-Balanced} та оцінену за набором даних CV.

Показник точності F1 для українсько-російської мовної пари показано на рис. 3.16 і 3.17.

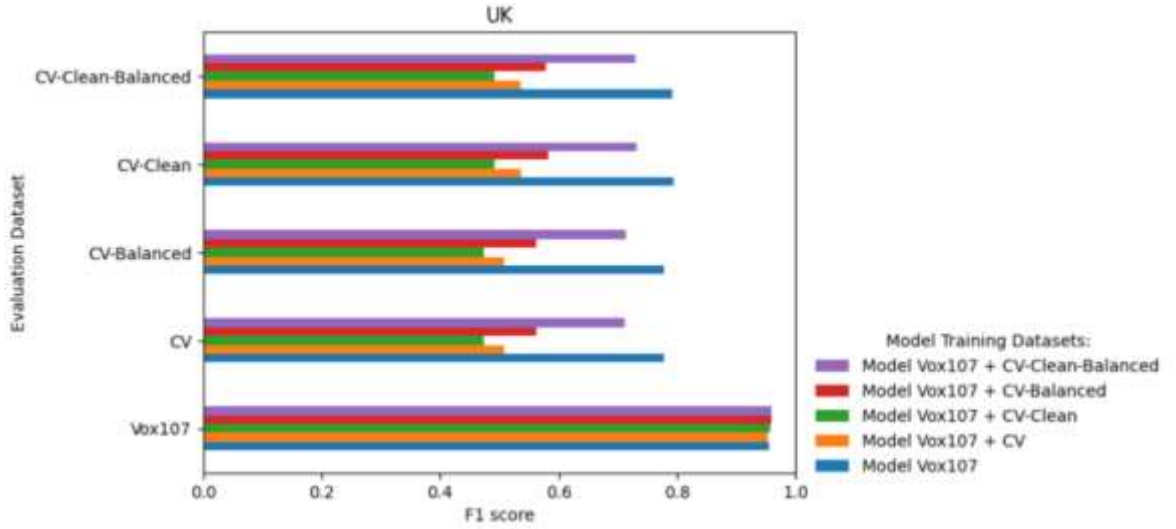


Рис. 3.16. Стовпчикова діаграма оцінки моделей для української мови

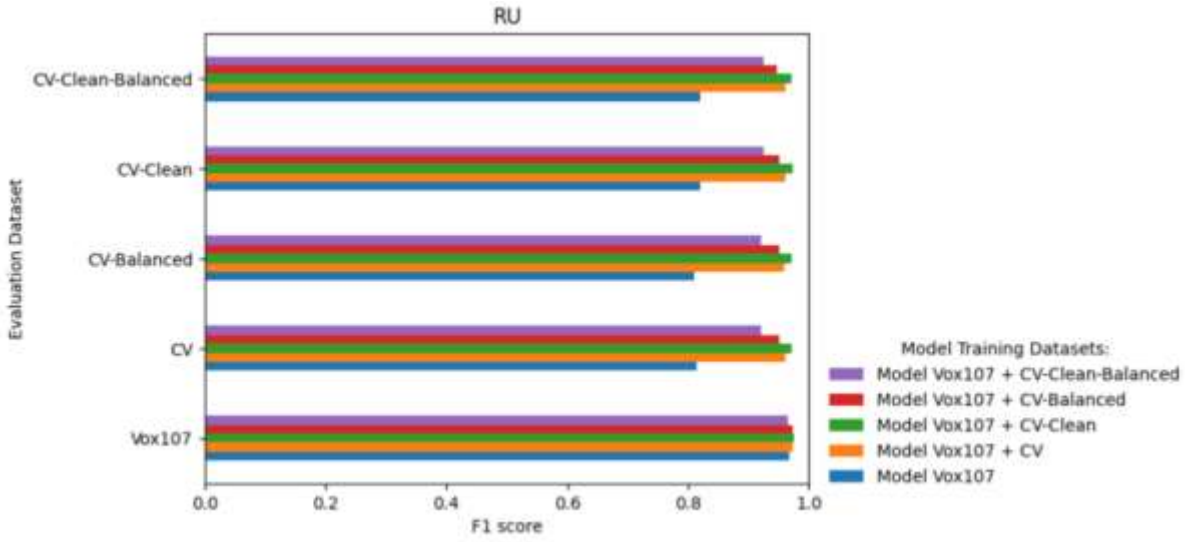


Рис. 3.17. Стовпчикова діаграма оцінки моделей для російської мови

Як видно з табл. 3.14, дані для України є обмеженими, а отже, значно незбалансованими порівняно з російським набором даних. Це призводить до погіршення точності для української мови в моделях, навчених на незбалансованих наборах даних. Значне середнє покращення було досягнуто під час навчання на CV-Clean-Balanced наборах даних.

3.4.6. Оцінка точності результатів для інших мовних пар

Поведінка чесько-словацької та шведсько-норвезької мовних пар схожа на українсько-російську, зі значним погіршенням точності для словацької та норвезької мов у моделях, навчених на незбалансованих наборах даних, але стабільними результатами після збалансування наборів даних. Після оцінювання моделей на новому наборі даних CV домену моделі, навчені на {VoxLingua107 + CV-Clean-Balanced}, показують на 5,8% кращі результати в середньому (і на 6,6% кращі в медіані) порівняно з моделями, навченими на {VoxLingua107}.

Порівнюючи якість та кількість, ми порівнюємо результати збалансованих моделей, навчених на наборі даних {VoxLingua107 + CV-Balanced} та {VoxLingua107 + CV-Clean-Balanced} (див. рис. 3.18). Основна відмінність полягає в порівнянні CV і CV-Clean частин навчальних даних для порівняння кількості (CV-Bal) і якості (CV-Cl-Bal).

Ми використовуємо два критерії для порівняння: Оцінка F1 та кількість даних. CV-Clean-Balanced показує на 1,4% кращі результати із середніми медіанами для всіх мов і використовує лише 30% обсягу даних для навчання порівняно з CV-Balanced.

Це означає втричі меншу кількість даних для досягнення кращої точності, що призводить до набагато кращої ефективності навчання, меншого часу навчання та позитивного впливу як з економічного, так і з екологічного боку за рахунок економії електрики на розрахунки.

Для досягнення значних покращень експерименти привели нас до наступних рекомендацій:

– тренуйте моделі SLI для конкретного регіону. Режим SLI для конкретного регіону перевершує багатомовну модель в середньому на 2,47% і в медіані на 1,65%, а також набагато швидше навчається і налаштовується;

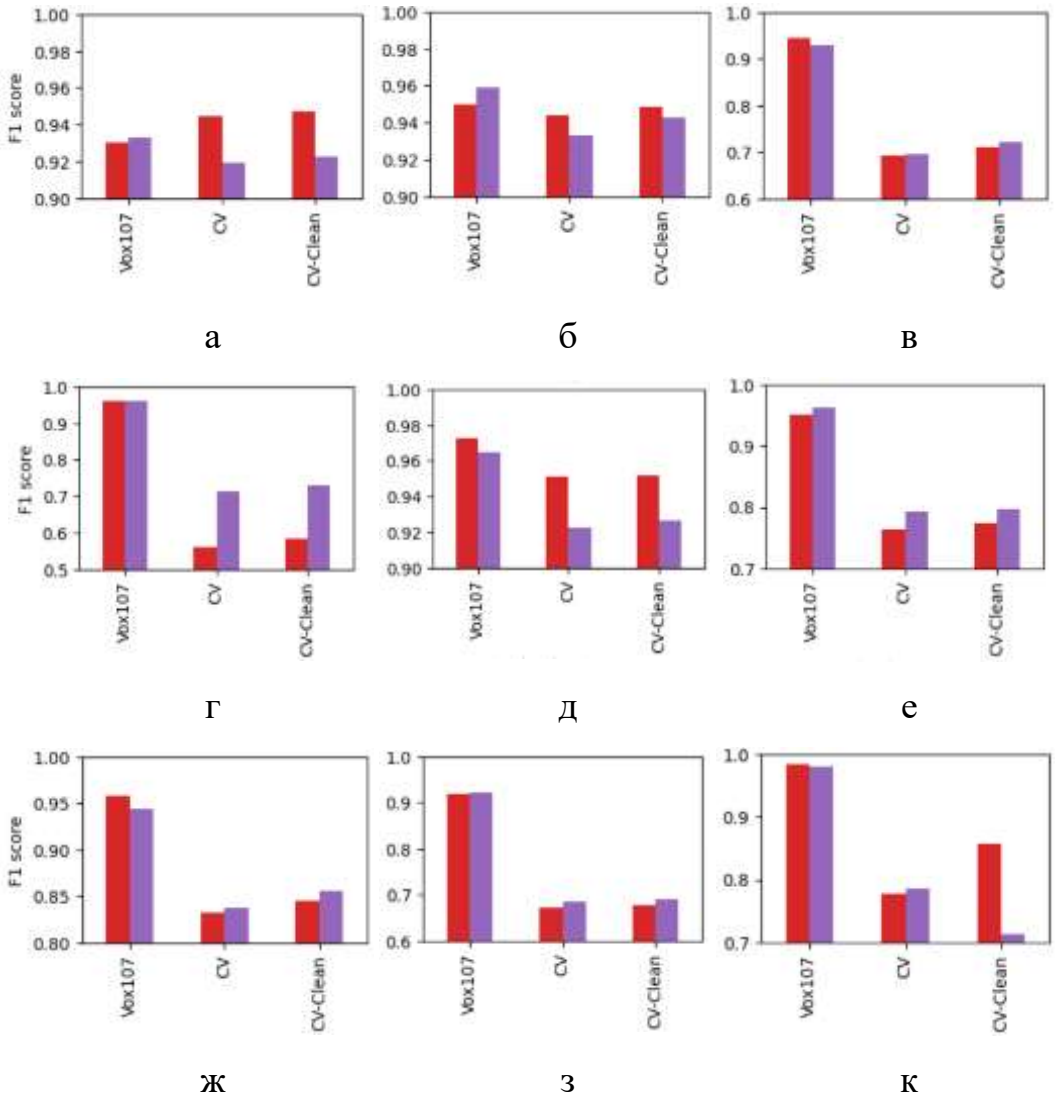


Рис. 3.18. Стовпчикові діаграми моделей, навчених на наборах даних {Vox107 + CV-Bal} та {Vox107 + CV-Cl-Bal} для (а) ES, (б) CA, (в) GL, (г) UK, (д) RU, (е) CS, (ж) SK, (з) SV та (к) NN

– додайте доменний набір даних. Моделі, навчені виключно на одному домені (набір даних VoxLingua107), дуже погано працюють на новому домені (CV). Додавання даних CV не зменшує точність, як це було оцінено на оригінальному VoxLingua107, що наводить нас на висновок, що якщо ми маємо дані домену, ми завжди повинні використовувати їх у навчанні з низьким впливом на вихідні дані домену. Погіршення точності за оцінкою на невидимому наборі даних CV в середньому становить 19,7%, а медіана – 17,6%;

– збалансувати набір даних. Як ми бачимо на прикладі шведсько-новонорвезьких пар і чесько-словацьких пар, а також іспансько-каталонсько-галісійських потрійних результатів, балансування відіграє вирішальну роль у досягненні високої точності для всіх мов. І це ще важливіше для невеликих ресурсних наборів даних, які містять лише десятки годин. Моделі, навчені на оригінальному та новому наборі даних {VoxLingua107 + CV-Clean-Balanced}, працюють на 5,8% краще в середньому (і на 6,6% краще в медіані) порівняно з моделями, навченими лише на оригінальному наборі даних VoxLingua107;

– очистити дані перед тренуванням.

Додавання лише 30% даних CV-Clean до тренінгу перевершило результати додавання всього набору даних CV на 1,4% із середніми значеннями для всіх мов, які брали участь у експерименті [43].

Висновки до розділу 3

1. В рамках дослідження сформульовані вимоги до даних для навчання мовних моделей та досліджені доступні мовні корпуси для української мови, що є основою для збору даних для навчання мовних моделей та розробки методів роботи із багатомовними аудіоданими. Для задачі сегментації речень підхід на основі маркування послідовностей значно перевершує підхід на основі мовного моделювання як за точністю оцінки F1, так і за часом прогнозування. Отримані дані були використані для проведення експериментальних даних в рамках апробації запропонованих методів підвищення якості та швидкості розпізнавання природної мови.

2. За допомогою отриманих маркованих аудіоданих перше був запропонований метод підвищення точності розпізнавання природної мови для близькоспоріднених мов, в якому при розпізнаванні природної мови фокус і увага були сконцентровані на точності, а вже в другу чергу на ширині покриття різних природніх мов, що дозволило вбудовувати розроблений метод в системи прийняття

рішень, підвищити точність роботи таких інформаційних систем в середньому на 19,7% (медіана 17,6%) і мінімізувати хибні спрацювання.

3. Під час дослідження також було покращено сегментацію неформатованого тексту з використанням мовного моделювання та маркування послідовностей, що дозволило зменшити об’єм потрібних для навчання мовної моделі аудіоданих та пришвидшити їхнє навчання. Слід зазначити, що китайська мова та набори даних є найскладнішими для перенесення на інші мови, переносимість англійських наборів даних, незважаючи на їхню доступність, є однією з найнижчих, отже, у виробничих умовах інженери не можуть покладатися на навчальні моделі англійською мовою для своєї мови.

4. В розділі був вдосконалений метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей, який на відміну від існуючих дозволяє використовувати розмічені на основі аудіоданих тексти та підвищити точність розпізнавання мови та зловмисних намірів від 29% до 94%.

5. Найважливішим етапом в забезпеченні швидкого виявлення на інцидентів є дослідження нових підходи до розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, а також способи підготовки та валідації вхідних аудіоданих. Таким чином, передбачення більшої кількості емоцій набагато складніше, тому рекомендується звузити набір даних до конкретного практичного використання і обмежити кількість емоцій, коли це можливо, замість того, щоб навчати модель на всьому наборі даних.

6. Запропоновані підходи до підвищення точності розпізнавання природної мови для близькоспоріднених мов вказує, що найвищої точності SLI-моделі можна досягти за рахунок кількох кроків: навчання SLI-моделі для конкретного регіону, додавання набору даних предметної області, збалансування набор даних і очищення даних перед навчанням. Всі ці кроки зроблять модель SLI набагато надійнішою для задач забезпечення безпеки голосової інформації.

7. В розділі представлений вдосконалений метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, який у порівнянні з

існуючими методами дає можливість більш точно визначати поріг емоційності для різних мов і тим самим мінімізувати нелегітимні спрацьовування на 18%. Даний метод набув подальшого розвитку за рахунок застосування його до забезпечення безпеки ІКС на підприємствах критичної інфраструктури та в державних органах.

8. Завдання обробки вибору мови із низької точністю для проведення експериментів та проведення тренінгу моделі в першу чергу дозволило верифікувати отримані результати експериментів. Також треба зазначити, що цілеспрямований підхід до тонкого налаштування з використанням кураторських наборів даних є більш ефективним, бо охоплює нюанси відмінностей між схожими мовами.

Список використаних джерел у розділі 3

1. 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 (PIC S&T)* (pp. 335–337). https://doi.org/10.1109/picst51311.2020.9468084

2. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Kipchuk, F., & Sukaylo, I. (2021). Automated Pipeline for Training Dataset Creation from Unlabeled Audios for Automatic Speech Recognition. *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 83, pp. 25–36). https://doi.org/10.1007/978-3-030-80472-5_3

3. Lyudovyk, T., & Pylypenko, V. (2014). Code-Switching Speech Recognition for Closely Related Languages. In *Workshop on Spoken Language Technologies for Under-Resourced* (pp. 1–6).

4. Lyudovyk, T., & Pylypenko, V. (2016). Bilingual Speech Recognition without Preliminary Language Identification (pp. 12–34).

5. Vasileva, N., Pilipenko, V., Radutsky, A., Robeyko, V., & Sazhok, N. (2012). Corpus of Ukrainian on-Air Speech. In *Speech Technology* (Vol. 2, pp. 12–21).

6. Meyer, J. (2020). Open Speech Corpora. https://github.com/JRMeyer/open-speech-corpora

7. MON Ukraine. (2021). YouTube. https://www.youtube.com/channel/UCQR9sMWcZshAwYX-EYH0qiA

8. Deutsche Welle in Ukrainian. (2021). YouTube. https://www.youtube.com/channel/UCQwVj4PyS5leCgEJY4I2t1Q

9. Toronto TV. (2021). YouTube. https://www.youtube.com/channel/UCF_ZiWz2Vcq1o5u5i1TT3Kw

10. Mozilla. (2021). Common Voice. https://commonvoice.mozilla.org/

11. TED. (2021). TED Talks. https://www.ted.com/talks

12. Iosifov, I., Iosifova, O., Sokolov, V., Skladannyi, P., & Sukaylo, I. (2022). Natural Language Technology to Ensure the Safety of Speech Information. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems II* (CPITS-II) (Vol. 3187(1), pp. 216–226).

13. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2002). Learning Representations by Back-Propagating Errors. In *Cognitive Modeling* (pp. 213–222). The MIT Press. https://doi.org/10.7551/mitpress/1888.003.0013

14. You, Y., & Nikolaou, M. (1993). Dynamic Process Modeling with Recurrent Neural Networks. In *AIChE Journal* (Vol. 39, no. 10, pp. 1654–1667). Wiley. https://doi.org/10.1002/aic.690391009

15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6000–6010).

16. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models Are Unsupervised Multitask Learners. *OpenAI* (Vol. 1, pp. 1–24).

17. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North. Association for Computational Linguistics* (pp. 1–16). https://doi.org/10.18653/v1/n19-1423

18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach (Version 1). *arXiv*. https://doi.org/10.48550/arXiv.1907.11692

19. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, *arXiv* (pp. 1–5). https://doi.org/10.48550/arXiv.1910.01108

20. Wikipedia. (2018). Wikipedia Monolingual Corpora. [Dataset]. https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/

21. Іосіфов, Є. (2023). Комплексний метод по автоматичному розпізнаванню природньої мови та емоційного стану. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 3(19), 146–164. https://doi.org/10.28925/2663-4023.2023.19.146164

22. ISO 639-6:2009. (2009). Codes for the Representation of Names of Languages. Part 6. Alpha-4 Code for Comprehensive Coverage of Language Variants, https://www.iso.org/standard/43380.html

23. Zhou, K., Sisman, B., Liu, R., & Li, H. (2021). Seen and Unseen Emotional Style Transfer for Voice Conversion with A New Emotional Speech Dataset. In *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. https://doi.org/10.1109/icassp39728.2021.9413391

24. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B. (2005). A Database of German Emotional Speech. In *Interspeech 2005*. ISCA. https://doi.org/10.21437/interspeech.2005-446

25. Pajupuu, H. (2012). Estonian Emotional Speech Corpus. Center of Estonian Language Resources. https://doi.org/10.15155/EKI.000A

26. Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., & Verma, R. (2014). CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset. In IEEE Transactions on Affective Computing (Vol. 5, no. 4, pp. 377–390). IEEE. https://doi.org/10.1109/taffc.2014.2336244

27. Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). IEMOCAP: Interactive Emotional Dyadic

Motion Capture Database. In *Language Resources and Evaluation* (Vol. 42, no. 4, pp. 335–359). Springer Science and Business Media LLC. https://doi.org/10.1007/s10579-008-9076-6

28. 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. In J. Najbauer (Ed.), *PLOS ONE* (Vol. 13, no. 5, p. e0196391). Public Library of Science (PLoS). https://doi.org/10.1371/journal.pone.0196391

29. Haq, S., & Jackson, P. J. B. (2011). Multimodal Emotion Recognition. In *Machine Audition* (pp. 398–423). IGI Global. https://doi.org/10.4018/978-1-61520-919-4.ch017

30. Pichora-Fuller, M. K., & Dupuis, K. (2020). Toronto Emotional Speech Set (TESS) [Dataset]. *Borealis*. https://doi.org/10.5683/SP2/E8H2MF

31. Mohamad Nezami, O., Jamshid Lou, P., & Karami, M. (2018). ShEMO: A Large-Scale Validated Database for Persian Speech Emotion Detection. In *Language Resources and Evaluation* (Vol. 53, no. 1, pp. 1–16). https://doi.org/10.1007/s10579-018-9427-x

32. Kerkeni, L., Cleder, C., Serrestou, Y., & Kosai Raoof. (2020). French Emotional Speech Database – Oréau (Version 1) [Dataset]. *Zenodo*. https://doi.org/10.5281/zenodo.4405783

33. Latif, S., Qayyum, A., Usman, M., & Qadir, J. (2018). Cross Lingual Speech Emotion Recognition: Urdu vs. Western Languages. In *2018 International Conference on Frontiers of Information Technology (FIT)*. IEEE. https://doi.org/10.1109/fit.2018.00023

34. Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN based Speaker Verification. In *Interspeech 2020*. ISCA. https://doi.org/10.21437/interspeech.2020-2650

35. Kumawat, P., & Routray, A. (2021). Applying TDNN Architectures for Analyzing Duration Dependencies on Speech Emotion Recognition. In *Interspeech 2021*. ISCA. https://doi.org/10.21437/interspeech.2021-2168

36. Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W.,

Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., De Mori, R., & Bengio, Y. (2021). SpeechBrain: A General-Purpose Speech Toolkit (Version 1). *arXiv*. https://doi.org/10.48550/arXiv.2106.04624

37. Iosifov, I., Iosifova, O., Romanovskyi, O., Sokolov, V., & Sukailo, I. (2022). Transferability Evaluation of Speech Emotion Recognition Between Different Languages. *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 134, pp. 413–426). https://doi.org/10.1007/978-3-031-04812-8_35

38. Koluguri, N. R., Park, T., & Ginsburg, B. (2021). TitaNet: Neural Model for Speaker Representation with 1D Depth-Wise Separable Convolutions and Global Context (Version 1). *arXiv*. https://doi.org/10.48550/arXiv.2110.04410

39. Jia, F., Koluguri, N. R., Balam, J., & Ginsburg, B. (2022). A Compact End-to-End Model with Local and Global Context for Spoken Language Identification (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.2210.15781

40. Kuchaiev, O., Li, J., Nguyen, H., Hrinchuk, O., Leary, R., Ginsburg, B., Kriman, S., Beliaev, S., Lavrukhin, V., Cook, J., Castonguay, P., Popova, M., Huang, J., & Cohen, J. M. (2019). NeMo: a toolkit for building AI applications using Neural Modules (Version 1). *arXiv*. https://doi.org/10.48550/arXiv.1909.09577

41. Valk, J., & Alumae, T. (2021). VoxLingua107: A Dataset for Spoken Language Recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. https://doi.org/10.1109/slt48900.2021.9383459

42. Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2019). Common Voice: A Massively-Multilingual Speech Corpus (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.1912.06670

43. Romanovskyi, O., Iosifov, I., Iosifova, O., Sokolov, V., Skladannyi, P., Sukaylo, I., & Tsarenok, O. (2024). Accuracy Improvement of Spoken Language Identification System for Close-related Languages. In Advances in Intelligent Systems, Computer Science and Digital Economics IV (pp. 1–18). [In press].

ВИСНОВКИ

У дисертації вирішено актуальне наукове завдання, яке полягає в підвищенні ефективності застосування методів та засобів забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації завдяки комбінуванню процесів підготовки навчальних аудіоданих та підходів до навчання та налаштування мовних моделей, якій застосовуються в ІКС. Дане наукове завдання має важливе значення для теорії і практики захисту інформації, створення та забезпечення функціонування інформаційних систем і технологій на об’єктах інформаційної діяльності та критичних інфраструктур сфери кібербезпеки та захисту інформації. Відсутність аналогічних рішень в нашій країні і закордоном робить результати досліджень пріоритетними.

Отримані результати мають важливе значення для модернізації існуючих та в процесі розробки нових методів захисту інформації.

На підставі проведених досліджень зроблені наступні висновки:

1. Вперше запропонований та математично обґрунтований метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів, який при навчанні на невеликій кількості нерозмічених даних дозволяє реалізувати підхід автоматичного отримання високоточного маркування, який дозволяє тренувати мовні моделі при наявності незначного обсягу маркованих аудіоданих (починаючи від 250 год.), що знижує вартість формування тренувального набору даних порівняно з ручним на 84% і пришвидшує процес маркуванням щонайменше на 85%, що в свою чергу знижує вартість тренування моделей на 61% і пришвидшує процес мінімум на 69%. Формалізована модель автоматизованого конвеєру дозволила створити навчальні набори даних з нерозмічених аудіозаписів та визначити критерії оцінки її роботи. Для цього був розроблений програмний код для автоматизованого створення навчальних наборів даних на основі нерозмічених аудіозаписів, що є обмеженням для навчання ASR-моделей для мов з низькими ресурсами та із специфічних доменів.

2. Вперше запропонований метод підвищення точності розпізнавання природної мови для близькоспоріднених мов, в якому при розпізнаванні природної

мови фокус і увага були сконцентровані на точності, а вже в другу чергу на ширині покриття різних природніх мов, що дозволило вбудовувати розроблений метод в системи прийняття рішень, підвищити точність роботи таких інформаційних систем в середньому на 19,7% (медіана 17,6%) і мінімізувати хибні спрацювання. Під час дослідження також було покращено сегментацію неформатованого тексту з використанням мовного моделювання та маркування послідовностей, що дозволило зменшити об’єм потрібних для навчання мовної моделі аудіоданих та пришвидшити їхнє навчання. Слід зазначити, що китайська мова та набори даних є найскладнішими для перенесення на інші мови, переносимість англійських наборів даних, незважаючи на їхню доступність, є однією з найнижчих, отже, у виробничих умовах інженери не можуть покладатися на навчальні моделі англійською мовою для своєї мови.

3. Вдосконалений метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей, який на відміну від існуючих дозволяє використовувати розмічені на основі аудіоданих тексти та підвищити точність розпізнавання мови та зловмисних намірів від 29% до 94%. Найважливішим етапом в забезпеченні швидкого визначення на інцидентів є дослідження нових підходи до розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, а також способи підготовки та валідації вхідних аудіоданих. Таким чином, передбачення більшої кількості емоцій набагато складніше, тому рекомендується звузити набір даних до конкретного практичного використання і обмежити кількість емоцій, коли це можливо, замість того, щоб навчати модель на всьому наборі даних.

4. Набув подальшого розвитку метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами, який у порівнянні з існуючими методами дає можливість більш точно визначати поріг емоційності для різних мов і тим самим мінімізувати нелегітимні спрацьовування на 18%. Даний метод набув подальшого розвитку за рахунок застосування його до забезпечення безпеки ІКС на підприємствах критичної інфраструктури та в державних органах. Завдання обробки вибору мови із низької точністю для проведення експериментів та

проведення тренінгу моделі в першу чергу дозволило верифікувати отримані результати експериментів. Також треба зазначити, що цілеспрямований підхід до тонкого налаштування з використанням кураторських наборів даних є більш ефективним, бо охоплює нюанси відмінностей між схожими мовами.

Мета дослідження щодо підвищення ефективності застосування безпечного розпізнавання та параметризації результатів обробки голосової інформації в ІКС завдяки комбінуванню підходів при формуванні розмічених аудіоданих для навчання мовних моделей досягнута і всі часткові завдання вирішені повністю. Наукові результати можуть бути використані дослідно-конструкторськими організаціями та державними структурами при розробці та удосконаленні систем ідентифікації загроз в аудіоінформації в режимі реального часу на об’єктах інформаційної діяльності критичної інфраструктури та державних органів. Перспективність запропонованих рішень для таких галузей як прихованої обробки аудіоданих, інтерактивних голосових меню для служб спасіння, медичних служб, банківської сфери, маркетплейсів, транскрибування телеконференцій тощо є очевидною.

В якості пріоритетних напрямів подальших досліджень планується проведення теоретичного пошуку методів підготовки даних, вдосконалення та пришвидшення методів розпізнавання та практичного підвищення захищеності аудіоданих, які передаються в державних та приватних ІКС.

Таким чином, поставлене актуальне наукове завдання розв’язане у повному обсязі. Усі визначені часткові завдання вирішено, мету досліджень досягнуто.

Додаток А

ПЕРЕЛІК МЕТОДІВ ENDERTURING SPEECH ENGINE ВЕРСІЇ 3.1.0929

Тип методу	Адреса методу	Опис методу
1	2	3
Логін ‘login’		
POST	/api/v1/login/access-token	Вхід за токеном, сумісним з OAuth2, отримання токену доступу для майбутніх запитів
POST	/api/v1/login/session-token	Зберегти токен, переданий у заголовку авторизації, лише в HTTP-файл cookie
DELETE	/api/v1/login/session-token	Видалити файл cookie токену сесії
POST	/api/v1/login/test-token	Тестовий токен доступу
POST	/api/v1/domain	Отримати домен
Агенти ‘agents’		
GET	/api/v1/agent-groups	Читати екземпляри
POST	/api/v1/agent-groups	Створити екземпляр
PUT	/api/v1/agent-groups/{pk}	Оновити екземпляр
DELETE	/api/v1/agent-groups/{pk}	Видалити екземпляр
GET	/api/v1/agents	Читання екземплярів
POST	/api/v1/agents	Створити екземпляр
DELETE	/api/v1/agents	Видалення екземплярів масово
PUT	/api/v1/agents/{pk}	Оновити екземпляр
DELETE	/api/v1/agents/{pk}	Видалити екземпляр
POST	/api/v1/agents/csv	Завантажити CSV
POST	/api/v1/agents/match-sessions	Зіставлення сеансів
POST	/api/v1/agents/crm-sync	Зв’яжіть агентів CRM з агентами SpeechEngine
GET	/api/v1/agents/experience-weights	Отримайте вагу досвіду
POST	/api/v1/agents/experience-weights/json	Завантажте вагу досвіду JSON
Анонімайзер ‘anonymizer’		
GET	/api/v1/anonymizer/languages	Отримати мову
PUT	/api/v1/anonymizer/languages/{language_code}	Оновити мову
ASR		
GET	/api/v1/asr	Читати екземпляри
GET	/api/v1/asr/languages	Отримати список мов. Мають бути показані лише активні

1	2	3
		(вже завантажені зображення) мови. Використовується для подання ASR
Автоматизація ‘automation’		
GET	/api/v1/automation	Читати завдання автоматизації
POST	/api/v1/automation	Створіть нове завдання автоматизації.
PUT	/api/v1/automation/{id}	Оновити екземпляр
GET	/api/v1/automation/{id}	Прочитати екземпляр
DELETE	/api/v1/automation/{id}	Видалити екземпляр
GET	/api/v1/automation/event-types	Отримати типи подій
POST	/api/v1/automation/{id}/run	Запустити завдання
Категорії ‘categories’		
GET	/api/v1/categories	Отримати категорії
POST	/api/v1/categories	Створити категорію
PUT	/api/v1/categories/positions	Оновити позицію категорії
PUT	/api/v1/categories/{pk}	Оновити категорію
DELETE	/api/v1/categories/{pk}	Видалити категорію
POST	/api/v1/categories/ get-sessions-number	Отримати кількість сесій для категорії. POST запит використовується, щоб уникнути проблем з довгими рядками запиту (фільтр може бути довшим за 2048 символів).
POST	/api/v1/categories/ rematch-sessions	Реванш сесій
Відповідність ‘compliance’		
GET	/api/v1/compliance/ script-adherence-rules	Отримати правила дотримання сценаріїв
POST	/api/v1/compliance/ script-adherence-rules	Створити правило дотримання сценарію
PUT	/api/v1/compliance/ script-adherence-rules/{rule_id}	Оновити правило дотримання сценаріїв
DELETE	/api/v1/compliance/ script-adherence-rules/{rule_id}	Видалити правило дотримання сценарію
Диск ‘drive’		
GET	/api/v1/drive/folders	Отримати папки
POST	/api/v1/drive/folders	Створити папку
PUT	/api/v1/drive/folders/{pk}	Оновити папку
DELETE	/api/v1/drive/folders/{pk}	Видалити папку
GET	/api/v1/drive/folders/{folder_id}/files	Отримати файли

1	2	3
GET	/api/v1/drive/files/{pk}	Отримати файл за ідентифікатором
PUT	/api/v1/drive/files/{pk}	Оновити файл
DELETE	/api/v1/drive/files/{pk}	Видалити файл
POST	/api/v1/drive/files	Створити файл
Файли ‘files’		
POST	/api/v1/files	Завантажити файл в систему
GET	/api/v1/files/{file_id}/download	Завантажити файл із системи
Фільтри ‘filters’		
GET	/api/v1/filters/favorite	Читати обране
POST	/api/v1/filters/favorite	Створити обране
PUT	/api/v1/filters/favorite/order	Оновити замовлення
PUT	/api/v1/filters/favorite/{pk}	Оновити обране
DELETE	/api/v1/filters/favorite/{pk}	Видалити обране
GET	/api/v1/filters/history	Читати історію
DELETE	/api/v1/filters/history/{pk}	Видалити історію
Воронка ‘funnel’		
GET	/api/v1/funnel	Отримати елементи
POST	/api/v1/funnel	Створити елемент
PUT	/api/v1/funnel/{id}	Оновити елемент
GET	/api/v1/funnel/{id}	Прочитати елемент
DELETE	/api/v1/funnel/{id}	Видалити позицію
GET	/api/v1/funnel/{id}/download	Завантажити XLSX
GET	/api/v1/funnel/download-step-contacts/{step_id}	Завантажити контакти кроку
Мітки ‘labels’		
GET	/api/v1/labels	Отримати мітки
POST	/api/v1/labels	Створити мітку
DELETE	/api/v1/labels/{label_id}	Видалити мітку
PUT	/api/v1/labels/{label_id}	Оновити мітку
POST	/api/v1/labels/{object_type}/{object_id}	Додати мітку до
Маркетплейс ‘marketplace’		
GET	/api/v1/marketplace/apps	Читати додатки
POST	/api/v1/marketplace/apps/{app_id}/ping	Додаток Ping
GET	/api/v1/marketplace/apps/{app_id}/icon	Завантажити іконку програми
PUT	/api/v1/marketplace/apps/{app_id}/oauth-code	Код встановлення OAuth
GET	/api/v1/marketplace/apps/{app_id}/oauth-url	Oauth Отримати URL-адресу
GET	/api/v1/marketplace/installations	Читати інсталяції

1	2	3
POST	/api/v1/marketplace/installations	Створити інсталяцію
GET	/api/v1/marketplace/installations/ {app_id}	Прочитати інсталяцію за ідентифікатором програми
PATCH	/api/v1/marketplace/installations/ {app_id}	Оновити інсталяцію
DELETE	/api/v1/marketplace/installations/ {app_id}	Видалити установку
PUT	/api/v1/marketplace/installations/ {app_id}/sync/{days}	Синхронізувати останні X днів
Сповіщення ‘notifications’		
GET	/api/v1/notifications	Отримувати сповіщення
GET	/api/v1/notifications/total	Отримати загальну суму
GET	/api/v1/notifications/settings	Налаштування
PUT	/api/v1/notifications/settings	Зберегти налаштування
GET	/api/v1/notifications/defaults	За замовчуванням
PATCH	/api/v1/notifications/read	Позначити як прочитане
Онбордінг ‘onboarding’		
GET	/api/v1/onboarding/groups	Читати групи
POST	/api/v1/onboarding/groups	Створити групу
POST	/api/v1/onboarding/groups/json	Імпорт групи з JSON
PUT	/api/v1/onboarding/groups/{pk}	Оновити групу
DELETE	/api/v1/onboarding/groups/{pk}	Видалити групу
POST	/api/v1/onboarding/groups/ {group_id}/steps	Створити крок
PUT	/api/v1/onboarding/groups/ {group_id}/steps/{step_id}	Оновити крок
DELETE	/api/v1/onboarding/groups/ {group_id}/steps/{step_id}	Видалити крок
PUT	/api/v1/onboarding/groups/ {group_id}/steps/{step_id}/ set-{new_status}	Встановити пройдений крок
Підказки ‘prompts’		
GET	/api/v1/prompts/models	Отримати доступні моделі
GET	/api/v1/prompts/categories/ {category}	Отримувати підказки
POST	/api/v1/prompts	Створити запит
DELETE	/api/v1/prompts/{prompt_id}	Видалити підказку
GET	/api/v1/prompts/{prompt_id}/ versions	Отримати версії підказок
PUT	/api/v1/prompts/{prompt_id}/ archived	Позначити як архівну версію
PUT	/api/v1/prompts/{prompt_id}/ versions/{row_id}/release	Позначити як версію

1	2	3
DELETE	/api/v1/prompts/{prompt_id}/ versions/{row_id}	Видалити версію
Розпізнавач ‘recognizer’		
POST	/api/v1/recognizer/audio-file	Завантажити аудіофайл
GET	/api/v1/recognizer/ filename-parser/{filename}	Парсер імен файлів
Звіти ‘reports’		
GET	/api/v1/reports/chart-data	Отримати дані діаграми
GET	/api/v1/reports/chart-data/ download/{file_format}	Завантажити дані діаграм
GET	/api/v1/reports/subscription	Прочитати підписку
PUT	/api/v1/reports/subscription	Створити або оновити підписку
POST	/api/v1/reports/subscription/send	Надіслати підписку
GET	/api/v1/reports/analytics-presets	Отримати всі аналітичні пресети
POST	/api/v1/reports/analytics-presets	Створити налаштування аналітики
PUT	/api/v1/reports/analytics-presets/{pk}	Оновити налаштування аналітики
DELETE	/api/v1/reports/analytics-presets/{pk}	Видалити налаштування аналітики
Оціночні картки ‘scorecards’		
GET	/api/v1/scorecards	Отримати список оцінок
POST	/api/v1/scorecards	Додати картку учасника
GET	/api/v1/scorecards/{pk}	Отримати картку учасника
PUT	/api/v1/scorecards/{pk}	Оновити картку учасника
DELETE	/api/v1/scorecards/{pk}	Видалити картку
GET	/api/v1/scorecards/{pk}/ sessions-number	Отримати кількість сеансів картки
POST	/api/v1/scorecards/{pk}/ archive	Архів
PUT	/api/v1/scorecards/{pk}/ copy	Копіювати екземпляр
POST	/api/v1/scorecards/{pk}/ points-dependencies	Отримати залежності балів
Сесії ‘sessions’		
GET	/api/v1/sessions	Читати сесії
DELETE	/api/v1/sessions	Видалення за фільтром
GET	/api/v1/sessions/discovery	Пошук сесій
GET	/api/v1/sessions/filter/choices	Прочитати вибір фільтрів
GET	/api/v1/sessions/filter/statistics	Зчитати фільтри Всього

1	2	3
GET	/api/v1/sessions/filter/ number_of_sessions	Отримати кількість сесій
POST	/api/v1/sessions/tasks/calculate-meta	Обчислити мета-дані
POST	/api/v1/sessions/tasks/retag	Відмітити сесії
POST	/api/v1/sessions/tasks/compliance/ match-script-adherence	Відповідність вимогам реваншу
POST	/api/v1/sessions/tasks/crm-sync	Синхронізація з CRM
POST	/api/v1/sessions/tasks/rescore	Переоцінка сесій
POST	/api/v1/sessions/chats	Завантаження чату
POST	/api/v1/sessions/loans	Завантаження кредиту
POST	/api/v1/sessions/manual	Завантаження вручну
GET	/api/v1/sessions/loans/brands	Список брендів кредитів
GET	/api/v1/sessions/{session_id}	Прочитати сесію за ідентифікатором
PUT	/api/v1/sessions/{session_id}	Оновити дані сесії
DELETE	/api/v1/sessions/{session_id}	Видалити
PUT	/api/v1/sessions/{session_id}/ metadata	Оновити метадані сеансу
PUT	/api/v1/sessions/{session_id}/ speaker	Перейменувати спікера
POST	/api/v1/sessions/{session_id}/ postprocess	Постобробка
POST	/api/v1/sessions/index	Перебудувати пошуковий індекс
POST	/api/v1/sessions/sync	Синхронізувати аналітичну базу даних
GET	/api/v1/sessions/{session_id}/ repetitives	Прочитати ланцюжок сеансів
GET	/api/v1/sessions/{session_id}/ compliance/script-adherence-matches	Читати збіги на відповідність
GET	/api/v1/sessions/{session_id}/ transcripts	Читання транскриптів
PUT	/api/v1/sessions/{session_id}/ transcripts/{transcript_id}	Оновити стенограму
GET	/api/v1/sessions/{session_id}/ file/{file_format}	Завантажити стенограму
GET	/api/v1/sessions/{session_id}/ transcripts-text	Отримати текст стенограми
GET	/api/v1/sessions/{session_id}/ comments	Отримати коментарі до сесії
POST	/api/v1/sessions/{session_id}/ comments	Додати коментар до сесії

1	2	3
PUT	/api/v1/sessions/{session_id}/ comments/{comment_id}	Редагувати коментар до сесії
DELETE	/api/v1/sessions/{session_id}/ comments/{comment_id}	Видалити коментар до заняття
GET	/api/v1/sessions/{session_id}/ scores	Отримати оцінку за сесію
POST	/api/v1/sessions/{session_id}/ scores	Встановити оцінку сесії
DELETE	/api/v1/sessions/{session_id}/ scores	Видалити оцінку сесії
GET	/api/v1/sessions/{session_id}/ default-scores	Отримати оцінку сесії за замовчуванням
POST	/api/v1/sessions/{session_id}/ scores/dispute	Оскаржити оцінку сесії
PUT	/api/v1/sessions/{session_id}/ scores/meta	Оновити мета-дані про оцінку сесії
GET	/api/v1/sessions/{session_id}/ scores/file	Завантажити оцінку сесії
GET	/api/v1/sessions/{session_id}/ scores/auto-qa-log	Отримати автоматичний журнал контролю якості
GET	/api/v1/sessions/{session_id}/ reactions	Отримати реакції на сесію
PUT	/api/v1/sessions/{session_id}/ reactions	Створити реакцію на сесію
GET	/api/v1/sessions/{session_id}/ summary	Отримати підсумок сесії
POST	/api/v1/sessions/{session_id}/ summary	Створити підсумок сесії
Налаштування ‘settings’		
GET	/api/v1/settings	Прочитати налаштування
PUT	/api/v1/settings	Оновити налаштування
Реєстрація ‘signup’		
POST	/api/v1/signup	Реєстрація
POST	/api/v1/signup/finish	Створити організацію з реєстраційних даних
Система ‘system’		
GET	/api/v1/system/license	Зчитати дані ліцензії
GET	/api/v1/system/license/limits-used	Зчитати використані ліміти читання
GET	/api/v1/system/timezones	Отримати часові пояси
Заміни ‘substitutions’		
GET	/api/v1/substitutions	Отримати екземпляри
POST	/api/v1/substitutions	Створити екземпляр

1	2	3
PUT	/api/v1/substitutions/{id}	Оновити екземпляр
DELETE	/api/v1/substitutions/{id}	Видалити екземпляр
POST	/api/v1/substitutions/json	Завантажити заміни JSON
Теги ‘tags’		
GET	/api/v1/tags	Отримати теги
POST	/api/v1/tags	Створити тег
POST	/api/v1/tags/json	Завантажити теги JSON
PUT	/api/v1/tags/do-tagging	Тегування
PUT	/api/v1/tags/{tag_id}	Оновити тег
DELETE	/api/v1/tags/{tag_id}	Видалити тег
POST	/api/v1/tags/{tag_id}/archive	Архів
Список справ ‘todo’		
GET	/api/v1/todo/own/{status}	Отримати елементи списку справ
POST	/api/v1/todo/excluded	Виключити завдання до списку справ
Тренінги ‘trainings’		
GET	/api/v1/trainings	Отримати тренінги
POST	/api/v1/trainings	Створити тренінг
DELETE	/api/v1/trainings/{id}	Видалити тренінг
GET	/api/v1/trainings/playlists	Отримати списки відтворення
POST	/api/v1/trainings/playlists	Створити плейлист
PUT	/api/v1/trainings/playlists/{pk}	Оновити список відтворення
DELETE	/api/v1/trainings/playlists/{pk}	Видалити список відтворення
GET	/api/v1/trainings/playlists/items	Отримати елементи списку відтворення сеансу
GET	/api/v1/trainings/playlists/ {playlist_id}/items	Отримати елементи списку відтворення
POST	/api/v1/trainings/playlists/ {playlist_id}/items	Створення елементів списку відтворення
PUT	/api/v1/trainings/playlists/ {playlist_id}/items/{item_id}	Оновити елемент списку відтворення
DELETE	/api/v1/trainings/playlists/ {playlist_id}/items/{item_id}	Видалити елемент списку відтворення
GET	/api/v1/trainings/quizzes/ users/me	Отримати мої тести
GET	/api/v1/trainings/quizzes/ users/me/summary	Отримати підсумок моїх тестів
GET	/api/v1/trainings/quizzes/ users/{user_id}/summary	Отримати підсумок вікторин агента
POST	/api/v1/trainings/quizzes/ users/me/{quiz_id}	Почати вікторину

1	2	3
PUT	/api/v1/trainings/quizzes/ users/me/{quiz_id}/results/ {result_id}	Оновити результат вікторини
GET	/api/v1/trainings/quizzes/ users/{user_id}/quiz/ {quiz_id}/results	Отримати результати тесту користувача
GET	/api/v1/trainings/quizzes	Отримати вікторини
POST	/api/v1/trainings/quizzes	Створити вікторину
PUT	/api/v1/trainings/quizzes/{pk}	Оновити вікторину
DELETE	/api/v1/trainings/quizzes/{pk}	Видалити вікторину
DELETE	/api/v1/trainings/quizzes/{pk}/stats	Видалити статистику вікторини
POST	/api/v1/trainings/quizzes/{pk}/copy	Копіювати вікторину
GET	/api/v1/trainings/skills	Отримати екземпляри
POST	/api/v1/trainings/skills	Створити екземпляр
PUT	/api/v1/trainings/skills/{pk}	Оновити екземпляр
DELETE	/api/v1/trainings/skills/{pk}	Видалити екземпляр
GET	/api/v1/trainings/plans	Отримати навчальні плани
POST	/api/v1/trainings/plans	Додати навчальний план
GET	/api/v1/trainings/plans/{plan_id}/ statistics	Отримати статистику за планом тренувань
PUT	/api/v1/trainings/plans/{plan_id}	Оновити навчальний план
DELETE	/api/v1/trainings/plans/{plan_id}	Видалити навчальний план
Користувачі ‘users’		
GET	/api/v1/users	Читати користувачів
POST	/api/v1/users	Створити користувача
POST	/api/v1/users/{user_id}/invite	Надіслати запрошення
GET	/api/v1/users/me	Прочитати мене
PUT	/api/v1/users/me	Оновити користувача Я
GET	/api/v1/users/me/notifications	Прочитати мої сповіщення
GET	/api/v1/users/suggestions	Отримати пропозиції користувачів
GET	/api/v1/users/roles	Отримати ролі
POST	/api/v1/users/roles	Створити роль
PUT	/api/v1/users/roles/{role_id}	Оновити роль
GET	/api/v1/users/{user_id}	Читати користувача за ідентифікатором
PUT	/api/v1/users/{user_id}	Оновити користувача

ДОДАТОК Б

ПРИКЛАД КЛАСУ ВЗАЄМОДІЇ З РОЗПІЗНАВАННЯ МОВЛЕННЯ

```
import json
import logging
import os

from pathlib import Path
from typing import Dict, List, Literal, Optional, Union
from urllib.parse import quote, urlencode, urlparse
from enderturing.config import AsrConfig, Config
from enderturing.ffmpeg_helper import _get_ffmpeg_file_cmd,
_get_ffmpeg_join_files_cmd
from enderturing.http_client import HttpClient
from enderturing.recognition_stream import RecognitionResultFormat,
RecognitionStream

log = logging.getLogger("enderturing")

class SpeechRecognizer:

  def __init__(self, config: Optional[AsrConfig] = None, api_config: Optional[Config] =
None):
    self.config = config or AsrConfig.from_env()
    self.api_config = api_config
    self._http_client = None

  def _build_server_url(self, **kwargs):
    return self.config.url + "?" + urlencode(kwargs, quote_via=quote)

  def stream_recognize_joined_file(
```

```
    self,
    channel_files: List[Union[str, Path]],
    *,
    result_format: Union[str, RecognitionResultFormat] =
RecognitionResultFormat.jsonl,
    include_partials: bool = False,
    extra_ws_params: dict = None,
  ):

    src_files = [Path(x) for x in channel_files]
    cmd, asr_channels = _get_ffmpeg_join_files_cmd(
      src_files, asr_sample_rate=self.config.sample_rate
    )
    res_format = result_format
    if isinstance(res_format, str):
      res_format = RecognitionResultFormat[res_format]
    asr_url = self._build_server_url(source="FILE", file_path=src_files[0].name)
    return RecognitionStream(
      asr_url=asr_url,
      cmd=cmd,
      src_file=src_files[0],
      asr_channels=asr_channels,
      sample_rate=self.config.sample_rate,
      res_format=res_format,
      extra_ws_params=extra_ws_params or {},
      include_partials=include_partials,
      max_ws_queue=self.config.max_ws_queue,
    )

  def stream_recognize(
```

```
    self,
    path: Union[str, Path],
    *,
    channels: Union[List[int], int, Literal["all", "mono"]] = "all",
    result_format: Union[
      Literal["text", "jsonl"], RecognitionResultFormat
    ] = RecognitionResultFormat.jsonl,
    include_partials: bool = False,
    extra_ws_params: dict = None,
  ) -> RecognitionStream:

    src_file = path if isinstance(path, Path) else Path(path)
    cmd, asr_channels = _get_ffmpeg_file_cmd(src_file, self.config.sample_rate,
channels)
    res_format = result_format
    if isinstance(res_format, str):
      res_format = RecognitionResultFormat[res_format]
    asr_url = self._build_server_url(source="FILE", file_path=src_file.name)
    return RecognitionStream(
      asr_url=asr_url,
      cmd=cmd,
      src_file=src_file,
      asr_channels=asr_channels,
      sample_rate=self.config.sample_rate,
      res_format=res_format,
      extra_ws_params=extra_ws_params or {},
      include_partials=include_partials,
      max_ws_queue=self.config.max_ws_queue,
    )
```

```
def _recognize_file(
  self,
  path: Union[str, Path],
  *,
  channels: Union[List[int], int, Literal["all", "mono"]] = "all", # TODO
  result_format: Union[
    Literal["text", "jsonl"], RecognitionResultFormat
  ] = RecognitionResultFormat.text,
  realtime: bool = False,
  api_config: Optional[Config] = None,
  speakers_names: Optional[Dict[int, str]] = None,
) -> dict:

  res_form = result_format
  if isinstance(res_form, str):
    res_form = RecognitionResultFormat[res_form]

  api_config = api_config or self.api_config
  if not api_config:
    raise ValueError("API config has to be set either in constructor or as argument")

  if isinstance(path, Path):
    path = str(path)

  parsed_path = urlparse(path)

  asr_token = self.config.url.split("/")[-1]
  params = dict(
    token=asr_token,
    language=self.config.language,
```

```
      caller_id=path,
      realtime=realtime,
      detailed_realtime_results=True if res_form is RecognitionResultFormat.jsonl else
False,
      channel_speaker_map=json.dumps(speakers_names),
    )

    if not self._http_client:
      # TODO instance per API config
      self._http_client = HttpClient(api_config)

    if parsed_path.scheme == "et":
      params["filename"] = parsed_path.path
      return self._http_client.post("/recognizer/local-audio-file", params=params)

    return self._http_client.post(
      "/recognizer/audio-file",
      params=params,
      files={"file": (os.path.basename(path), open(path, "rb"), "audio/mpeg")},
    )

  def recognize_file(
    self,
    path: Union[str, Path],
    *,
    result_format: Union[
      Literal["text", "jsonl"], RecognitionResultFormat
    ] = RecognitionResultFormat.text,
    api_config: Optional[Config] = None,
  ) -> dict:
```

```
  return self._recognize_file(
    path, realtime=True, result_format=result_format, api_config=api_config
  )

def recognize_file_long(
  self,
  path: Union[str, Path],
  *,
  api_config: Optional[Config] = None,
  speakers_names: Optional[Dict[int, str]] = None,
) -> dict:

  return self._recognize_file(
    path, realtime=False, api_config=api_config, speakers_names=speakers_names
  )
```

ДОДАТОК В

ПРИКЛАДИ КЛАСІВ ДЛЯ РОЗПІЗНАВАННЯ АУДІОДАНИХ

```
import asyncio
import json
import logging
import socket
import ssl
import websockets

from asyncio import Event
from collections import deque
from enum import Enum, unique
from io import TextIOBase
from websockets.connection import State

log = logging.getLogger("enderturing")

@unique
class RecognitionResultFormat(Enum):
 jsonl = 1
 text = 2

class RecognitionStream(TextIOBase):
 def __init__(
  self,
  *,
  asr_url,
  cmd,
  src_file,
  asr_channels,
```

```
 sample_rate,
 extra_ws_params,
 res_format,
 include_partials,
 max_ws_queue
):
 self._asr_url = asr_url
 self._src_file = src_file
 self._cmd = cmd
 self._asr_channels = asr_channels
 self._sample_rate = sample_rate
 self._extra_ws_params = extra_ws_params or {}
 self._res_format = res_format
 self._include_partials = include_partials
 self._max_ws_queue = max_ws_queue
 self._tracker = None
 self._websocket = None
 self._finished = False
 self._buffer = deque()

def readable(self) -> bool:
 """Marks that the stream is readable."""
 return True

def seekable(self) -> bool:
 """Marks that the stream is not seekable."""
 return False

def writable(self) -> bool:
 """Marks that the stream is not writable."""
```

```
 return False

async def read(self, size=-1) -> str:
 """Reads an entire transcript."""
 ready = []
 ready_len = 0
 max_len = size if size and size > 0 else 1e9
 while ready_len < max_len and not self._finished:
  if len(self._buffer) == 0:
   event = Event()
   self._tracker["buffer_waiter"] = event
   await event.wait()
  if len(self._buffer) > 0:
   row = self._buffer.popleft()
   ready_len += len(row) + 1
   ready.append(row)
 return "\n".join(ready)

async def readline(self, size=-1):
 """Reads a next transcript line."""
 if len(self._buffer) == 0 and not self._finished:
  event = Event()
  self._tracker["buffer_waiter"] = event
  await event.wait()
 if self._finished and len(self._buffer) == 0:
  return ""
 return self._buffer.popleft()

async def _close(self) -> None:
 if self._websocket and self._websocket.state == State.OPEN:
```

```
  await self._websocket.close()
 # cancel any pending tasks
 for task in [self._tracker["task_reader"], self._tracker["task_sender"]]:
  if not task.done():
   task.cancel()
  else:
   task.result()

async def __aexit__(self, exc_type, exc, tb):
 await self._close()

async def _ws_reader(
 self, websocket, result_format: RecognitionResultFormat, include_partials: bool,
tracker
):
 try:
  while True:
   response_json = json.loads(await websocket.recv())
   if "ts" in response_json:
    tracker["received"] = response_json["ts"]
    if (
     tracker["waiter"]
     and tracker["sent"] - tracker["received"] < self._max_ws_queue
    ):
     event = tracker["waiter"]
     tracker["waiter"] = None
     event.set()
   if log.isEnabledFor(logging.DEBUG):
    log.debug(json.dumps(response_json, ensure_ascii=False))
   if "text" in response_json.keys() and response_json["text"]:
```

```
    if result_format == RecognitionResultFormat.text:
     self._buffer.append(response_json["text"])
    elif result_format == RecognitionResultFormat.jsonl:
     self._buffer.append(json.dumps(response_json, ensure_ascii=False))
   elif include_partials and result_format == RecognitionResultFormat.jsonl:
    self._buffer.append(json.dumps(response_json, ensure_ascii=False))
   if len(self._buffer) and tracker["buffer_waiter"]:
    event = tracker["buffer_waiter"]
    tracker["buffer_waiter"] = None
    event.set()
 except websockets.ConnectionClosed as e:
  log.info("WS read connection closed for %s: %s", str(self._src_file), str(e))
 except Exception as e:
  log.error("WS read error for %s: %s", str(self._src_file), str(e))
 finally:
  self._finished = True
  if tracker["buffer_waiter"]:
   event = tracker["buffer_waiter"]
   tracker["buffer_waiter"] = None
   event.set()

async def _file_sender(self, websocket, cmd, num_channels, tracker):
 proc = await asyncio.create_subprocess_exec(*cmd, stdout=asyncio.subprocess.PIPE)
 sample_rate = self._sample_rate
 try:
  while True:
   data = await proc.stdout.read(sample_rate)
   if len(data) == 0:
    await websocket.send('{"eof" : 1}')
    break
```

```
   await websocket.send(data)
   tracker["sent"] += len(data) / num_channels / sample_rate / 2
   if tracker["sent"] - tracker["received"] >= self._max_ws_queue:
    event = Event()
    tracker["waiter"] = event
    await event.wait()
  except websockets.ConnectionClosed as e:
   log.info("WS write connection closed for %s: %s", str(self._src_file), str(e))
  except Exception as e:
   log.error("WS write error for %s: %s", str(self._src_file), str(e))

 def _get_ssl_ctx(self, asr_url: str):
  if asr_url.startswith("wss"):
   ssl_ctx = ssl.create_default_context()
   ssl_ctx.check_hostname = False
   ssl_ctx.verify_mode = ssl.CERT_NONE
  else:
   ssl_ctx = None
  return ssl_ctx

 async def __aenter__(self):
  log.info("Connecting to: '%s'", self._asr_url)
  try:
   websocket = await websockets.connect(
    self._asr_url, ssl=self._get_ssl_ctx(self._asr_url)
   )
  except socket.gaierror as e:
   log.error("Error during connection to WS: %s", e)
   log.error("Make sure host accessible and DNS records exists for: '%s'", self._asr_url)
   raise
```

```
await websocket.send(
 json.dumps(
  {
   "config": {
    "sample_rate": self._sample_rate,
    "channels": self._asr_channels,
    **self._extra_ws_params,
   }
  }
 )
)
self._tracker = {
 "sent": 0,
 "received": 0,
 "waiter": None,
 "buffer_waiter": None,
}
self._websocket = websocket
self._tracker["task_sender"] = asyncio.create_task(
 self._file_sender(websocket, self._cmd, self._asr_channels, self._tracker)
)
self._tracker["task_reader"] = asyncio.create_task(
 self._ws_reader(websocket, self._res_format, self._include_partials, self._tracker)
)
return self
```

ДОДАТОК Г

ПРИКЛАДИ КЛАСІВ ДЛЯ FFMPEG МАНІПУЛЯЦІЇ

```
import json
import logging
import subprocess
from pathlib import Path
from typing import List, Literal, Union

log = logging.getLogger("enderturing")

_ffmpeg_defaults = ["ffmpeg", "-nostdin", "-loglevel", "quiet"]

def _exec_cmd(cmd):
  p = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
  out, err = p.communicate()
  if p.returncode != 0:
    raise RuntimeError(cmd[0], out, err)
  return out.decode("utf-8")

def get_num_channels(path: Path):
  cmd = ["ffprobe", "-show_format", "-show_streams", "-of", "json", path]
  res = json.loads(_exec_cmd(cmd))
  num_channels = int(res["streams"][0]["channels"])
  log.debug("Identified %d channels in '%s'", num_channels, str(Path))
  return num_channels

def _get_ffmpeg_mono_file_cmd(path: Path, *, asr_sample_rate: int):
  cmd = [*_ffmpeg_defaults, "-i", str(path), "-ar", str(asr_sample_rate), "-ac", "1", "-f",
"s16le", "-"]
  return cmd
```

```
def _get_ffmpeg_file_cmd_channels_list(path: Path, *, asr_sample_rate: int, channels:
Union[List[int], int] = None):
  cmd = [*_ffmpeg_defaults, "-i", str(path)]
  if channels:
    dst = channels if isinstance(channels, list) else range(channels)
    for channel in dst:
      cmd.append("-map_channel")
      cmd.append(f"0.0.{channel}")
  cmd.extend(["-ar", str(asr_sample_rate), "-f", "s16le", "-"])
  return cmd

def _get_ffmpeg_join_files_cmd(files: List[Path], *, asr_sample_rate: int):
  cmd = _ffmpeg_defaults.copy()
  merge_filter = ""
  for idx, path in enumerate(files):
    cmd.append("-i")
    cmd.append(str(path))
    merge_filter += f"[{idx}:a]"
  merge_filter += f"amerge=inputs={len(files)}[a]"
  cmd.extend(["-filter_complex", merge_filter, "-map", "[a]", "-ar", str(asr_sample_rate),
"-f", "s16le", "-"])
  return cmd, len(files)

def _get_ffmpeg_file_cmd(path: Path, sample_rate: int, channels: Union[List[int], int,
Literal["all", "mono"]] = "all"):
  if channels == "mono":
    asr_channels = 1
    cmd = _get_ffmpeg_mono_file_cmd(path, asr_sample_rate=sample_rate)
    return cmd, asr_channels
```

```
  detected_channels = get_num_channels(path)
  if channels == "all":
    asr_channels = detected_channels
    cmd = _get_ffmpeg_file_cmd_channels_list(path, asr_sample_rate=sample_rate)
    return cmd, asr_channels

  if isinstance(channels, list):
    valid_channels = [channel for channel in channels if channel < detected_channels]
    if len(valid_channels) < len(channels):
      log.warning(
        "Some values in `channels` are out of range, ignored %d provided values",
        len(channels) - len(valid_channels),
      )
    asr_channels = len(valid_channels)
    cmd = _get_ffmpeg_file_cmd_channels_list(path, asr_sample_rate=sample_rate,
channels=valid_channels)
    return cmd, asr_channels

  if isinstance(channels, int):
    if channels > detected_channels:
      log.warning(
        "Requested %d channels for recognition, but file '%s' has %d channels, using lower
value",
        channels,
      )
      asr_channels = detected_channels
    else:
      asr_channels = channels
    if asr_channels == detected_channels:
      cmd = _get_ffmpeg_file_cmd_channels_list(path, asr_sample_rate=sample_rate)
```

```
    else:
      cmd = _get_ffmpeg_file_cmd_channels_list(path, asr_sample_rate=sample_rate,
channels=asr_channels)
    return cmd, asr_channels

  raise ValueError("Unsupported `channels` value")
```

ДОДАТОК Д

ПРИКЛАДИ КЛАСІВ ДЛЯ РОБОТИ ІЗ HTTP-ПОТОКОМ

```
import datetime
import logging
import requests
import urllib3

from typing import Optional, TypedDict, Union

logger = logging.getLogger("enderturing")

class AuthData(TypedDict):
  token_type: str
  access_token: str
  expires_in: int
  expires_on: datetime.datetime

TOKEN_EXP_TIME_WINDOW = datetime.timedelta(seconds=60)

class HttpClient:
  def __init__(self, config):
    self.config = config
    self._auth_data: Optional[AuthData] = None
    if not config.ssl_verify:
      urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

  def _get_full_api_url(self, path) -> str:
    return f"{self.config.url.strip('/')}/api/{self.config.api_version}{path}"
```

```
  def _get_auth_data(self) -> AuthData:
    if self._auth_data:
      if self._auth_data["expires_on"] - TOKEN_EXP_TIME_WINDOW <
datetime.datetime.now():
        logger.info("Token expired, refreshing...")
        self._auth_data = None
      else:
        return self._auth_data

    auth_data = {"username": self.config.auth_key, "password": self.config.auth_secret}
    logger.info("Authenticating to Ender Turing SpeechEngine: %s", self.config.url)
    auth = requests.post(
      self._get_full_api_url("/login/access-token"),
      data=auth_data,
      verify=self.config.ssl_verify,
    )
    auth.raise_for_status()
    authorization_json = auth.json()

    self._auth_data = AuthData(
      token_type=authorization_json["token_type"],
      access_token=authorization_json["access_token"],
      expires_in=int(authorization_json["expires_in"]),
      expires_on=datetime.datetime.now()
      + datetime.timedelta(seconds=int(authorization_json["expires_in"])),
    )

    logger.info(
      "Authenticated, got token: xxxxx%s (expires in: %s seconds)",
      self._auth_data["access_token"][-4:],
```

```
    authorization_json["expires_in"],
  )
  return self._auth_data

def _get_auth_headers(self) -> dict:
  data = self._get_auth_data()
  return {"Authorization": f"{data['token_type']} {data['access_token']}"}

def get(self, url: str, params: Optional[dict] = None) -> Union[dict, list, str, bytes]:
  """Executes authorized GET requests to API."""
  response = requests.get(
    self._get_full_api_url(url),
    headers=self._get_auth_headers(),
    params=params,
    verify=self.config.ssl_verify,
  )
  response.raise_for_status()
  if response.headers["content-type"] == "application/json":
    response_json = response.json()
    logger.debug("JSON response for %s: %s", url, str(response_json))
    return response_json
  elif response.headers["content-type"].startswith("text"):
    logger.debug(
      "text response (%s) for %s: %s",
      response.headers["content-type"],
      url,
      response.text,
    )
    return response.text
```

```
  else:
    logger.debug(
      "bytes response (%s) for %s: %s...",
      response.headers["content-type"],
      url,
      response.content[:1000],
    )
    return response.content

def put(self, url: str, json: Union[dict, list] = None, **kwargs) -> Union[dict, list]:
  """Executes authorized PUT requests to API."""
  response = requests.put(
    self._get_full_api_url(url),
    headers=self._get_auth_headers(),
    json=json,
    verify=self.config.ssl_verify,
    **kwargs,
  )
  response.raise_for_status()
  response_json = response.json()
  logger.debug("JSON response for %s: %s", url, str(response_json))
  return response_json

def post(self, url: str, json: Union[dict, list] = None, **kwargs) -> dict:
  """Executes authorized POST requests to API."""
  response = requests.post(
    self._get_full_api_url(url),
    headers=self._get_auth_headers(),
    json=json,
    verify=self.config.ssl_verify,
```

```
    **kwargs,
  )
  response.raise_for_status()
  response_json = response.json()
  logger.debug("JSON response for %s: %s", url, str(response_json))
  return response_json

def delete(self, url: str, json: Union[dict, list] = None, **kwargs) -> Union[dict, list]:
  """Executes authorized DELETE requests to API."""
  response = requests.delete(
    self._get_full_api_url(url),
    headers=self._get_auth_headers(),
    json=json,
    verify=self.config.ssl_verify,
    **kwargs,
  )
  response.raise_for_status()
  response_json = response.json()
  logger.debug("JSON response for %s: %s", url, str(response_json))
  return response_json
```

Додаток Е

АКТ ВПРОВАДЖЕННЯ В КИЇВСЬКОМУ СТОЛИЧНОМУ УНІВЕРСИТЕТІ ІМЕНІ БОРИСА ГРІНЧЕНКА

КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОРИСА ГРІНЧЕНКА

ФАКУЛЬТЕТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ТА МАТЕМАТИКИ
вул. Лєвка Лук'яненка, 13-Б, м. Київ, Україна, 04207
Тел.: +380 44 428-34-14
fitm.kubg.edu.ua, fitm@kubg.edu.ua

BORYS GRINCHENKO
KYIV METROPOLITAN UNIVERSITY

FACULTY
OF INFORMATION TECHNOLOGIES
AND MATHEMATICS
13-B Levka Lukianenka St. Kyiv, Ukraine, 04207
Tel.: +380 44 428-34-14
fitm.kubg.edu.ua, fitm@kubg.edu.ua

27.08.2024 № 19

АКТ
про впровадження результатів дисертаційного дослідження
Іосіфова Євгена Анатолійовича
на тему «Методи та засоби забезпечення безпечного розпізнавання та параметризації результатів обробки голосової інформації», поданої на здобуття наукового ступеня доктора філософії зі спеціальності 125 Кібербезпека

Цим Актом, грунтуючись на рішенні кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка, засвідчуємо, що нижчеперелічені наукові положення, а саме:

1) метод автоматизованого конвеєру для створення навчальних наборів даних з нерозмічених аудіозаписів;
2) метод підвищення точності розпізнавання розмовної мови для близькоспоріднених мов;
3) метод сегментації неформатованого тексту з використанням мовного моделювання та маркування послідовностей;
4) метод розпізнавання багатомовних емоцій шляхом оцінки переносу між різними мовами

розроблені особисто Іосіфовим Євгеном Анатолійовичем у ході проведення ним дисертаційних досліджень та отримали високу оцінку при обговоренні на засіданнях кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка.

Зазначені наукові результати:

по-перше, впроваджені в освітній процес кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка, що відображено у програмах навчальних дисциплін спеціальності 125 Кібербезпека за захист інформації першого (бакалаврського), другого (магістерського) та третього (освітньо-наукового) рівнів вищої освіти;

по-друге, впроваджені в програмно-апаратне забезпечення лабораторій безпеки інформаційних активів, антивірусного захисту інформації, систем технічного та криптографічного захисту інформації.

Дослідження Іосіфова Євгена Анатолійовича відповідає всім вимогам до організації наукового пошуку та дає позитивний результат у практичному застосуванні.

Декан
Факультету інформаційних технологій та математики
кандидат фізико-математичних наук,
старший науковий співробітник Оксана ЛИТВИН

Додаток Ж

АКТ ВПРОВАДЖЕННЯ В ENDER TURING OÜ

ENDER TURING

Ender Turing OÜ

Ender Turing OÜ
No. 534-24
09/07/2024

Certificate

on the implementation of the results of the dissertation research by **Ievgen Iosifov**
on the topic: **"Methods and Means of Ensuring Secure Recognition and Parameterization of Speech Information Processing Results"**

This certifies that the scientific results of the dissertation research of Ievgen Iosifov for the degree of Doctor of Philosophy in the specialty "Cybersecurity" were used in the work of Ender Turing OÜ.

- The proposed method of an automated pipeline for creating training datasets from unlabeled audio recordings is valuable and noteworthy for the company. Due to the critical shortage of labeled audio data, the implemented approach to automatically obtaining high-precision labeling can significantly save computing resources and execution time.
- The improved method of segmentation of unformatted text using modeling and sequence labeling allows the use of labeled texts based on audio data and thus increases the efficiency of speech and malicious intent recognition subsystems.

The results of the study have practical applications for improving the performance of algorithms for processing streaming audio information and ensuring its integrity and availability.

Olena Iosifova

Management board member,

Ender Turing OÜ

Додаток І

АКТ ВПРОВАДЖЕННЯ В PP 2 SPV LIMITED LIABILITY COMPANY

PP 2 SPV SP Z OO
UL. MARKA KOTAŃSKIEGO 1
10-166 OLSZTYN
17.07.2024

PP 2 SPV Limited Liability Company,
ul. Marka Kotańskiego 1, 10-166 Olsztyn, Poland
KRS 0000927630, NIP 7393959736, REGON 520218860

CERTIFICATE

on the implementation of the results obtained during the dissertation research
by **Ievgen Iosifov**
on the topic: "**Methods and Means of Ensuring Secure Recognition and Parameterization of Speech Information Processing Results**"

This certificate indicates that the results obtained in the dissertation research of Ievgen Iosifov were used in the course of the work of PP 2 SPV Limited Liability Company to increase the level of security of audio information by determining the emotional state of the subscriber in real-time while simultaneously recognizing several languages or their mixture.

List of realization and implementation of the research results:

- For the first time, a method for improving the accuracy of spoken language recognition for closely related languages is proposed, focusing on accuracy, unlike existing recognition approaches that focus on wider language coverage. This makes it possible to integrate the developed method into decision-making systems in which the accuracy of spoken language detection affects their further analysis, which increases the accuracy of such systems and minimizes false positives.

- The method of recognizing multilingual emotions was improved by assessing the transfer between different languages, which, together with the method of spoken language recognition, makes it possible to more accurately determine the threshold of emotionality for different languages and thereby minimize illegitimate alarms, including the level of natural emotionality of individual peoples, which allowed to calibrate the data for the implementation of security measures at the state level.

The scientific and practical results of the study can find further practical application in the process of developing and improving existing mechanisms for securely recognizing the results of processing voice information transmitted within an information system.

Olena Iosifova,
Member of the Management Board