КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

Дослідження підходів машинного навчання синтезу мови

Виконала студентка 4-го курсу
Кузьмяк Анна Юріївна _____ (підпис)

Науковий керівник:
професор, доктор фіз.-мат. наук _____ (підпис)
Крак Юрій Васильович

Засвідчую, що в цій курсовій
роботі немає запозичень з праць
інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної кібернетики
«____» _____ 2022 р.,
протокол № ___

Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК _____
(підпис)

КИЇВ-2022

РЕФЕРАТ

Обсяг роботи 40 сторінок, 13 ілюстрацій, 19 джерел посилань.

Ключові слова: СИНТЕЗ МОВИ, МАШИННЕ НАВЧАННЯ, СПЕКТРОГРАМА, ВОКОРДЕР, ГРАДІЄНТНИЙ СПУСК, ФУНКЦІЯ АКТИВАЦІЇ, НОРМАЛІЗАЦІЯ, НЕЙРОННІ МЕРЕЖІ, SEQ2SEQ, ФОНЕМА, CNN, RNN, GAN, МЕХАНІЗМ УВАГИ

Об’єктом дослідження є підходи машинного навчання у процесі синтезу мови - конвертування тексту у мовний сигнал.

Метою роботи є аналіз і порівняння підходів машинного навчання у синтезі мови на різних його етапах.

Результати роботи: досліджені сучасні підходи машинного навчання у задачі синтезу мови, а саме проведено детальний аналіз та порівняння технологій та шляхів розв’язання проблем, що стоять на різних етапах синтезу мови.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

CNN – convolutional neural network, згорткова нейронна мережа

RNN – recurrent neural network, рекурентна нейронна мережа

DNN – deep neural network, глибока нейронна мережа

АЧХ – Амплітудно-частотна характеристика

TTS – Text-to-Speech, текст-у-мовлення

AI – Artificial Intelligence, штучний інтелект

IVR – Interactive voice response, інтерактивна голосова відповідь

MOS – Mean opinion score, середня оцінка думки

Seq2seq – Sequence to sequence

LSTM – Long short-term memory, довга короткочасна пам'ять

G2P – grapheme-to-phoneme

ВСТУП

Оцінка сучасного стану об’єкта дослідження. Задача синтезу мови з’явилась відносно нещодавно і вже має низку підходів для її вирішення. Нові, сучасні технології розробляються швидко, відбувається покращення наявних підходів на різних етапах синтезу мови та їх адаптація під різні умови використання, тож важливо розуміти, як працюють та якими бувають ці підходи, а також розуміти їхні переваги та недоліки.

Актуальність роботи та підстави для її виконання. На сьогодні синтез мови застосовується у різних сферах та для різних цілей – голосові помічники, розумні будинки, системи голосових меню(IVR-системи), допомога людям з вадами зору, як-от, читачі екрану, та загалом будь-які системи, де людина отримує інформацію від комп’ютера. З розвитком нейронних мереж та машинного навчання вдалось значно просунутись в цьому напрямку та покращити якість синтезованого мовлення.

Оскільки відбувається активний розвиток різних технологій і підходів у напрямку застосування машинного навчання у синтезі мови, потрібно розуміти фундаментальні основи та принципи, покладені в роботу цих підходів, їх об’єктивні переваги та недоліки, щоб мати повну картину про сучасні технології машинного навчання та їх застосування у розв’язанні задачі синтезу мови, адже ця задача, як і багато інших, має свої особливості. Задача складніша, ніж може здатись на перший погляд, адже крім основних вимог – розбірливість та натуральність звучання, є й інші, такі як підтримка багатомовності, використання інтонацій та виразності, що адаптуються під різні сфери застосування, підтримка різних частин мови, як-от абревіатури, числівники, омографи тощо.

Мета й завдання роботи. Метою роботи є дослідження і порівняння підходів машинного навчання у синтезі мови на різних його етапах.

Для досягнення цієї мети були поставлені такі завдання:

- Дослідити процес синтезу мову, його особливості та основні етапи.
- Проаналізувати підходи для вирішення задач синтезу мови на різних етапах, зокрема підходи машинного навчання, які допомогли значно покращити технологію синтезу мови.

1. ЗАДАЧА СИНТЕЗУ МОВЛЕННЯ

Синтез мови, або Text-to-Speech(TTS) – це моделювання людського мовлення з текстового представлення зазвичай через методи машинного навчання. Зазвичай синтез мови використовують для створення інтерактивних голосових роботів, наприклад, для IVR(системи голосових меню). Основна вимога до згенерованого мовлення – натуральність його звучання, для цього потрібно врахувати роботу над тембром, плавністю, розстановкою акцентів та пауз, інтонацій тощо, та його зрозумілість. Побудова будь-якої системи синтезу мовлення передбачає збір даних, але кількість необхідних даних залежить від підходу.

1.1. ПІДХОДИ СИНТЕЗУ МОВЛЕННЯ

Існує два підходи до цього процесу: конкатенативний та параметричний. Конкатенативний підхід(unit selection) – склеювання фрагментів записаного аудіо. Таке мовлення має високу якість, але потребує велику кількість даних для машинного навчання. Параметричний підхід – побудова ймовірнісної моделі, яка обирає акустичні властивості звукового сигналу для заданого тексту.

Мовлення моделей unit selection має високу якість, низьку варіативність та потребує великого обсягу даних для навчання. У той же час для тренування параметричних моделей необхідна набагато менша кількість даних, вони генерують різноманітніші інтонації, але донедавна страждали від загальної досить низької якості звуку порівняно з підходом unit selection.

Проте з розвитком технологій глибокого навчання моделі параметричного синтезу досягли суттєвого приросту за всіма метриками

якості і здатні створювати мовлення, що практично не відрізняється від людського.

Перевагами підходу unit selection є природність звуку, висока швидкість генерації. Недоліками є те, що синтезована мова монотонна, не містить емоцій, вимагає досить великої тренувальної бази аудіоданих для покриття різноманітних контекстів, у принципі не може генерувати звук, який не зустрічається в навчальній вибірці та характерні артефакти склеювання.

Перевагами параметричного синтезу є природне та плавне звучання при використанні end-to-end підходу, більша різноманітність в інтонаціях, використання меншого обсягу даних, порівняно з моделями unit selection. Недоліки це низька швидкість роботи проти unit selection та велика обчислювальна складність.

1.2. ЕТАПИ СИНТЕЗУ МОВИ

Незважаючи на велику кількість досліджень, сучасні системи синтезу мови влаштовані досить складно. Щоб конвертувати текст у мовлення, система машинного навчання повинна пройти кілька етапів. Спочатку алгоритм повинен конвертувати текст у формат, який можна прочитати, тобто здійснити нормалізацію. На цьому етапі потрібно врахувати числа, дати, абревіатури у тексті тощо. Алгоритм поділяє текст на окремі фрази, які система читає з певною інтонацією. Під час цього програма дотримується певної пунктуації та структури у тексті. Оскільки процес нормалізації не дуже простий та однозначний, так як потребує розуміння контексту, для цієї задачі часто використовують нейронні мережі. Також необхідно мати словник наголосів, або ж враховувати правила мови, якщо це можливо.

Після потрібно проаналізувати просодичні особливості тексту, а саме виділити фрази, щоб розділяти їх паузами, а також визначити інтонацію.

Для розуміння правильної вимови система використовує вбудовані словники. Якщо потрібне слово відсутнє, алгоритм створює транскрипцію, використовуючи загальні правила. Для кожного слова повинен бути список фонем, що входять до складу слова. Альтернатива цьому – розбити слова на графеми (письмові складові одиниці, які зазвичай складаються з окремих букв або складів), а потім генерувати фонеми, які відповідають їм, використовуючи набір правил.

Далі потрібно згенерувати звук – на цьому етапі алгоритм називається вокодер. Найчастіше у системах синтезу мовлення текст спочатку перетворюється у спектральний вигляд – спектрограму, а потім у безпосередньо голос. Звук розбивають на відрізки з деяким кроком та за допомогою перетворення Фур'є для кожного відрізку рахують спектр і представляють у вигляді графіка, який показує час та частоту(рис. 1)

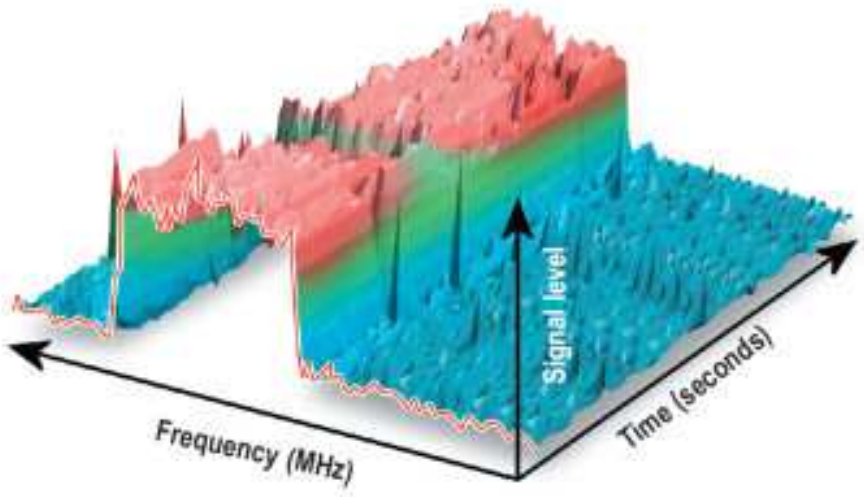

Рисунок 1 – Спектрограма

Якщо частоту виразити не в герцах, а в мелах, така спектограма називатиметься мел-спектограмою – часто саме таке представлення використовують в алгоритмах синтезу мови.

Далі потрібно з сформованої спектограми відновити звук. Для цього існують різні способи, такі як, наприклад, алгоритм Гріффіна-Ліма, який

дозволяє наближено відтворити звук за його амплітудним спектром, або його більш сучасні модифікації. Та чинником суттєвого приросту якості синтезованої мови стало застосування саме нейромережевих вокодерів замість алгоритмів цифрової обробки сигналів, наприклад відома реалізація WaveNet, яка поступово передбачає значення амплітуди звукової хвилі.

Узагальнена схема синтезу мовлення поетапно зображена на рисунку 2.

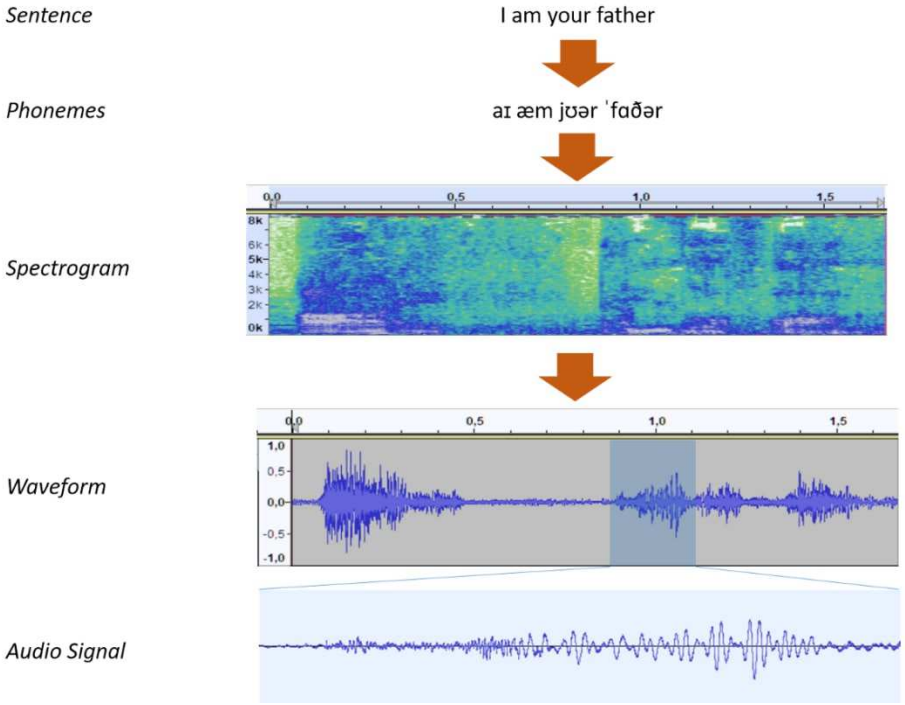

Рисунок 2 – Етапи синтезу мовлення

Отже, на кожному етапі синтезу мови застосовують підходи машинного навчання, про які буде далі йти мова.

2. ЗАГАЛЬНІ ВІДОМОСТІ ПРО МАШИННЕ НАВЧАННЯ

Машинне навчання – це галузь комп'ютерної науки, алгоритми якої покращуються автоматично на основі вхідних даних та досвіду. Основна ідея машинного навчання – навчання на основі даних, ідентифікація певних патернів та прийняття рішення з мінімальним втручанням людини.

Як і інші галузі комп’ютерних наук, машинне навчання значно розвинулось за останні роки. Народилось воно з ідеї, що комп’ютер може виконувати певні задачі без того, щоб програмувати конкретні алгоритми, які задають виконання цих задач. Важливим також є таке навчання, що при появі нових даних модель може обробити їх та сама адаптуватись. Таке самостійне навчання, в тому числі на власних помилках дає можливість приймати у достатній мірі надійні рішення та результати.

Можливість автоматично застосовувати складні математичні обрахунки на великий даних достатньо швидко з’явилась відносно нещодавно. Такі речі як зростання кількості та різноманітності доступних даних, обчислювальні потужності, які стали дешевшими та ефективнішими та більш доступне зберігання даних зробили цю галузь важливою та актуальною.

Машинне навчання використовується для вирішення задач у різних галузях. У логістиці воно застосовується для оптимізації логістичних маршрутів, прогнозування попиту та годин пік та загального покращення продуктивності ланцюгів постачання. У галузі охорони здоров’я значне використання машинного навчання у носимих девайсах та сенсорах, які у режимі реального часу аналізують показники здоров’я людини, а також у медицині для допомоги постановки діагнозів та назначення лікування. Також машинне навчання застосовується для забезпечення безпеки, протидії шахрайству та ідентифікації злочинців. У торгівлі та маркетингу аналіз великих даних дозволяє платформам для продажу аналізувати попередні

покупки та персоналізувати рекомендації та рекламу, впроваджувати маркетингові кампанії, оптимізацію цін та планування збуту. У фінансовій галузі машинне навчання застосовується для автоматизації консультативних послуг для інвесторів, попередження шахрайства, для алгоритмічного трейдингу.

Виділяють такі методи машинного навчання, як навчання з учителем, напівавтоматичне навчання, навчання з підкріпленням та навчання без учителя. Алгоритми машинного навчання з учителем тренуються на прикладах з мітками, наприклад, інформація на вході, де бажана інформація на виході нам відома. Через такі методи як класифікація, регресія, передбачення, градієнтне підсилення навчання з учителем використовує патерни, щоб передбачити значення міток на додаткових даних, де цих міток немає. Це потребує того, щоб алгоритм навчання узагальнював навчальні дані достатньо розумним способом, щоб застосувати ці узагальнення на тих даних, з якими він ще не знайомий. Навчання з учителем часто застосовується там, де історичні дані з достатньою ймовірністю передбачають майбутні події.

Напівавтоматичне навчання має таке ж застосування, що і навчання з учителем, але використовує як дані з мітками, так і без них для навчання – зазвичай невелику кількість даних з мітками, і велику кількість без, оскільки дані без міток більш доступні та не такі дорогі. Таке навчання може застосовуватись з такими методами, як класифікація, регресія та передбачення. Напівавтоматичне навчання є корисним тоді, коли вартість промаркованих занадто висока, щоб застосовувати навчання з учителем.

Навчання з підкріпленням часто застосовується у робототехніці, іграх, навігації. Алгоритми такого навчання виявляють через спроби та помилки, які дії призводять до найбільших винагород. За відсутності навчального датасету, доводиться навчатись на основі власного досвіду. Навчання з підкріпленням

має три основні компоненти: агент, що приймає рішення, середовище – все, з чим взаємодіє агент, та дії – усе, що агент може робити. Мета агента – обрати дії для максимізації очікуваної нагороди за заданий період часу. Математичною основою для опису середовища у навчанні з підкріпленням є марковський процес вирішування, за допомогою якого можна сформулювати майже всі задачі навчання з підкріпленням.

Навчання без учителя використовує немарковані дані. Системі не говорять, які відповіді є правильними, натомість алгоритм сам повинен дослідити дані та знайти певні закономірності та структуру. Навчання без учителя добре працює на даних про транзакції, наприклад, щоб ідентифікувати клієнтів з схожими атрибутами, які можна трактувати схожим чином під час маркетингових кампаній. Розповсюдженими техніками у навчанні без учителя є самоорганізаційна карти Кохонена, кластеризація методом k-середніх, метод головних компонент, сингулярне представлення матриці тощо.

3. НЕЙРОННІ МЕРЕЖІ ТА ЇХ ТИПИ

Нейронні мережі – це набір алгоритмів, змодельованих за зразком людського мозку, що спроектовані для розпізнавання шаблонів. Вони інтерпретують дані через машинне сприйняття, маркування та кластеризацію вхідних даних. Патерни, які вони розпізнають, мають вигляд числових векторів, у які повинні бути перетворені дані реального світу, як-от зображення, звук, текст або часові ряди.

Нейронні мережі допомагають у кластеризації та класифікації, у групуванні немаркованих даних відповідно до подібностей серед вхідних даних, а також класифікують дані, коли є промаркований набір даних, на якому можна навчатись.

Глибоким навчанням називають нейронні мережі, які складаються з більше ніж трьох шарів. Шари складаються з вузлів – це ті місця, де відбуваються обчислення. Вузол комбінує вхідні дані з набором коефіцієнтів, або вагів, які підсилюють, або послаблюють вхідні дані, призначаючи даним певну ступінь важливості відповідно до задачі алгоритму. Значення на виході сумуються, і тоді сума проходить через функцію активації для визначення того, в якій мірі сигнал повинен впливати на кінцевий результат(рисунок 3). Вихідні дані кожного шару є одночасно вхідними даними для наступного шару починаючи від початкового шару, який отримує вхідні дані. Чим більш просунутою є нейронна мережа, тим більш комплексні ознаки можуть розпізнавати вузли, оскільки вони об’єднують та рекомбінують ознаки з попередніх шарів. Це дозволяє нейронним мережам глибокого навчання обробляти дійсно дуже великі набори даних з мільярдами параметрів.

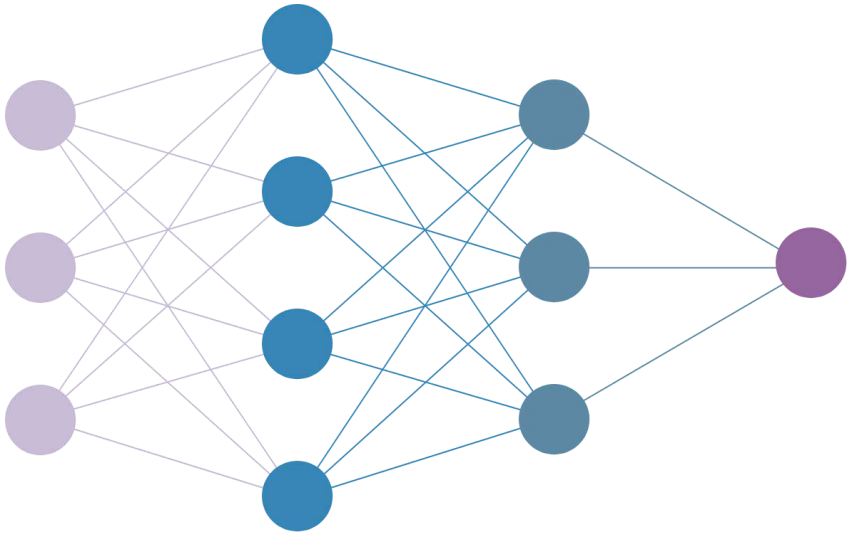

Рисунок 3 – Схематичне зображення нейронної мережі

На даний момент для навчання нейронних мереж майже усюди використовується алгоритм зворотного поширення помилки. Результат обчислення на переданій множині прикладів звіряється з очікуваним результатом (розміченими даними). Різниця між фактичним та очікуваним значеннями називають помилкою та поширюють цю помилку на ваги мережі у зворотному напрямку. Таким чином, мережа адаптується під розмічені дані і, як правило, результат цієї адаптації добре працює і для тих даних, які мережа не зустрічала у вихідних прикладах для навчання.

Що не менш важливо, нейронні мережі здатні знаходити приховану структуру в неструктурованих даних, якими є більшість даних у реальному світі. Відповідно, одна з проблем, яку глибоке навчання вирішує найкраще це обробка та кластеризація необроблених даних, медіа, виявлення подібностей у даних, які люди жодним чином не впорядкували. Мережі глибокого навчання виконують автоматичне вилучення ознак без людського втручання, на відміну від більш традиційних методів машинного навчання.

Різні архітектури нейронних мереж розроблені для роботи з різними типами даних та у різних доменних областях. Найпростішим та найстарішим типом є перцептрон, який складається з одного нейрону, який застосовує

функцію активації до вхідних даних, щоб отримати дані на виході. Він не містить жодних прихованих шарів і може використовуватись у завданнях бінарної класифікації.

Нейронна мережа прямого поширення складається з нейронів та прихованих шарів, які пов’язані між собою. Рух даних відбувається лише у прямому напрямку, дані не передаються назад. Що більша кількість шарів, то більші можливості для налаштування вагів, і відповідно, здатності навчатись. Ваги не оновлюються, оскільки немає зворотного поширення. Такі нейронні мережі застосовують у класифікації, розпізнаванні мови, патернів, облич. Найголовнішим недоліком таких мереж є їхня нездатність навчатись через зворотне поширення помилки.

Багатошаровий перцептрон - нейронна мережа, яка поєднує кілька прихованих шарів та функції активації.Тут ваги оновлюються за допомогою градієнтного спуску. Багатошаровий перцептрон є двонаправленим, у напрямку вперед поширюються вхідні дані, у зворотному – ваги.

Мережа радіально базисних функцій використовує інший спосіб для передбачення. Вона складається з вхідного шару, шару з нейронами радіальних базисних функцій та вихідних даних. Нейрони зберігають фактичні класи для кожного екземпляру навчальних даних. Як функція активації використовується радіальна функція.

Для класифікації зображень найчастіше використовують згорткові нейронні мережі(CNN). Згорткові нейронні мережі містять згорткові шари, які відповідають за вилучення важливих ознак з зображення. Операція згортки використовує матрицю, так званий фільтр, який ініціалізується випадковим чином та змінюється шляхом зворотного поширення помилки та формує карту фільтру. Одним з прикладів такого фільтру є оператор Кенні для виділення границь у зображеннях. Після згорткового агрегувальний шар, який відповідає за агрегацію карт, сформованих на попередньому шарі.

Згорткові мережі використовують ReLU(зрізаний лінійний вузол) як функцію активації у прихованих шарах. Останнім шаром є повноз'єднаний шар, який зазвичай використовує Softmax для класифікації та ReLU для регресії як функції активації.

Рекурентні нейронні мережі(RNN) застосовують, коли є необхідність у передбаченнях з використанням послідовних даних – послідовностей зображень, слів тощо. Нейронна мережа відрізняється від мережі прямого поширення тим, що попередні дані можуть бути використані як вхідні, при цьому маючи приховані стани. На відміну від найпростіших багатошарових перцептронів, рекурентні мережі здатні використовувати внутрішню пам'ять для обробки послідовностей довільної довжини. Недоліком є проблема зникнення градієнту.

Довга короткочасна пам'ять(LSTM) – архітектура рекурентних нейронних мереж, що здатна вирішити проблему зникнення градієнту шляхом додавання клітинок пам’яті, які зберігають інформацію на досить довгий період. Блоки мережі містять три вентилі: вхідний, вихідний і забувальний для визначення того, які вихідні дані потрібно зберегти або забути. Вхідний вентиль контролює, які дані потрібно тримати в пам’яті, вихідний контролює, які дані передавати у наступний шар, забувальний – коли відкинути дані, які не є потрібними. LSTM архітектура нейронних мереж набула широкого поширення в завданнях обробки текстів через те, що вона здатна обробляти тексти довільної довжини та аналізувати контекст.

4. ОГЛЯД ПРОЦЕСУ ТА ЕТАПІВ СИНТЕЗУ МОВИ

4.1. ОБРОБКА ТА НОРМАЛІЗАЦІЯ ТЕКСТУ

Виконанню синтезу мови слідує попередній етап підготовки тексту – нормалізація. Тут тексту потрібно надати зручний для озвучки вигляд. Це включає у себе:

- Запис дат та чисел
- Розшифрування абревіатур, скорочень, спеціальних символів
- Обробка елементів алфавіту або слів, іншою мовою – їх транскрипція

На перший погляд такі задачі можуть здаватись тривіальними. Але основною проблемою, що виникає при вирішенні таких задач є неоднозначність в прочитанні елементів. Часто скорочення, дати, числа тощо не завжди можна однозначно перетворити – існує залежність від контексту, в якому вони використовуються. Простий приклад: текст “2-і” можна перетворити як на порядковий числівник “другі”, так і на кількісний “дві”. Це ж стосується і форми, у якій стоїть числівник – без контексту не можна однозначно сказати, у якому відмінку, роді та числі потрібно поставити слово. Варто додати, що нормалізація дуже залежить від мови, наприклад, у англійській слова змінюються не так як в українській, отже для синтезу англійської мови може бути достатньо звичайних засобів нормалізації, що не включають машинне навчання. Значною проблемою в українській є контекст є вибір форми, у якій повинно стояти слово.

Для задачі нормалізації з урахуванням контексту добре підходить архітектура нейронної мережі Sequence to Sequence (seq2seq) – вона приймає на вхід послідовність і генерує іншу послідовність. Seq2seq мережі отримали значне застосування у завданнях машинного перекладу, сумаризації тексту

тощо. Такі моделі побудовані з двох пов’язаних RNN – кодувальник та декодувальник. Кодувальник обробляє кожен елемент вхідної послідовності, переводить отриману інформацію у вектор чисел з рухомою комою, що називається контекстом. Після обробки всієї вхідної послідовності кодувальник пересилає контекст декодувальнику, який потім починає генерувати вихідну послідовність елемент за елементом(рисунок 4).

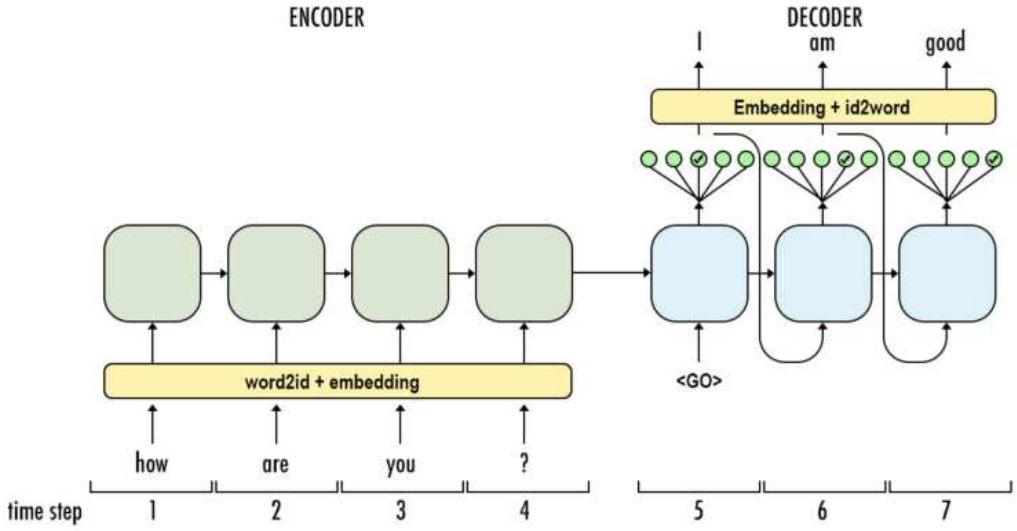

Рисунок 4 – Схема роботи seq2seq моделі

При навчанні моделі можна задати розмір контекстного вектора – число прихованих нейронів (hidden units) в декодувальнику. Для перетворення слова у вектор використовують ряд алгоритмів, що називаються вкладання слів(word embeddings). Вони представляють слова у векторні простори, які містять семантичну інформацію про них. Можна використовувати як уже навчені ембеддінги, так і навчити ембеддінг на своєму наборі даних. Перші моделі такі як Word2Vec «розуміли» зміст тексту лише на рівні окремих слів, без контексту. Останні досягнення у комп'ютерній лінгвістиці дозволили перейти до ефективних векторних уявлень для цілих речень та абзаців тексту.

На вхід декодувальнику на вихідному такті подається зарезервований символ start of string, на кожному наступному такті подається прихований

стан з попереднього кроку і згенероване в попередню ітерацію слово. Генерація відповіді триває доти, доки не буде згенеровано спеціальне слово кінця рядка end of string.

Такого типу моделям складно мати справу з довгими реченнями – ця проблема вирішується за допомогою техніки, що називається “механізм уваги”. Механізм уваги додатково вводиться між кодером і декодером для того, щоб знайти, на якому вихідному представленні зосередитися під час прогнозування поточного елемента, і що є важливим компонентом для навчання послідовності. Сигнали уваги формуються таким чином. Внутрішні стани кодувальника обробляються деякою нейронною архітектурою, що повертає вектор вагів для оброблених станів. Далі сума станів надходить на вхід декодувальнику і використовується нарівні з виходом кодувальника. На кожному кроці роботи декодувальника сигнал оновлюється з урахуванням попереднього внутрішнього стану декодувальника. Така архітектура дозволяє нейронній мережі вивчити «важливість» фрагментів послідовності, що надходить на вхід і точніше передбачати результат.

Також потрібно підготувати словник наголосів. Розстановка наголосів залежить від мови синтезу: вона може виконуватись за правилами мови, з певною кількістю винятків, а може базуватись на словнику, якщо таких правил немає.

Наступний етап – просодична обробка. Просодична обробка тексту полягає у наданні тексту інтонаційного оформлення. Сюди належить поділ тексту на просодичні одиниці – синтагми, визначення довжини пауз між синтагмами і вибір інтонації для кожної з них. Синтагма – головна одиниця реалізації інтонації, що характеризується інтонаційною та смисловою цілісністю та акцентно-ритмічною структурою. Просодія в більшості покладається на синтаксис. Хоч на неї впливає і семантика, але оскільки в

даний час доступно мало даних про генеративні аспекти цієї залежності, системи синтезу мови часто зосереджені саме на синтаксисі.

Межі синтагми можуть маркуватися паузами, всередині синтагми паузи неприпустимі. У складі синтагми виділяється головне слово, що отримує синтагматичний наголос. Такі наголоси можна проставляти на основі певних правил мови. Недоліки такого підходу – він не враховує винятки та більш складні випадки, а також потребує розробки нових правил для нової мови.

4.2. СИНТЕЗ СПЕКТРОГРАМИ

Далі текст розбивають на фонеми. Процесс перетворення тексту в фонеми може відбуватись на основі двох підходів — словарного і заснованого на правилах. Словниковий підхід використовує словник із записаними фонетичними представленнями слів і в процесі роботи здійснює пошук у ньому з метою конвертації слова в послідовність фонем. Заснований на правилах підхід використовує набір правил, що застосовуються до слів з метою виділення фонем. Обидва ці підходи мають недоліки та потребують удосконалення. Для поділу тексту на фонеми часто використовують G2P(grapheme-to-phoneme) – перетворення графем на фонеми для генерації фонетичної транскрипції з письмового вигляду. G2P перетворення вивчається протягом тривалого часу. Системи G2P, засновані на правилах, використовують широкий набір правил для конвертації у фонеми. Розробка такої системи G2P вимагає лінгвістичного досвіду. Крім того, деякі мови (наприклад, китайська та японська) мають складні системи письма, а створення правил є трудомістким і важко охопити більшість можливих ситуацій.

Останнім часом для G2P почали застосовуватися нейронні мережі. Перетворення G2P на основі нейронної мережі є надійним проти

орфографічних помилок, можна легко інтегрувати в end-to-end системи синтезу мови, які повністю побудовані з глибоких нейронних мереж. В більшості G2P моделі побудовані на основі seq2seq архітектруи. Існують різні реалізації моделей на основі LSTM або CNN, які мають різну ефективність. Так, наприклад, аналіз показав, що високу ефективність демонструють моделі з CNN як кодувальника, і LSTM як декодувальника зі сторони меншої кількості помилок. Модель з тільки згортковими шарами є швидшою за інші моделі, та все ж має непогану точність.

Приклад кінцевого результату перетворення слова у фонеми в українській мові: результатом транскрипції слова “козацька” є послідовність ['K', 'AO', 'Z', 'AA1', 'TSJ', 'K', 'AA'], або ж у більш зрозумілому людині вигляді, [к о з á ц’ к а].

Наступним етапом генерують спектрограму. Перетворення Фур'є розкладає функцію часу на частоти. Воно відображає амплітуду кожної частоти, присутньої в базовій функції, що можна побачити на спектрограмі(рис. 5). Існує альтернативне тривимірне уявлення, яке може допомогти зрозуміти спектрограму(рис. 1). Отже, спектрограма – двовимірна математична модель звуку, яка у найпростішому вигляді представляє декартову площина, в якій вісь X представляє час, а вісь Y — частоту.

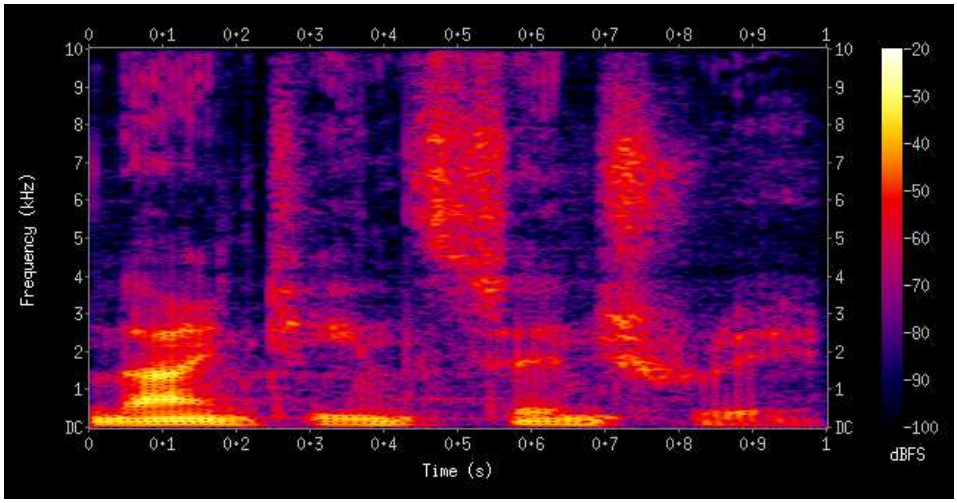

Рисунок 5 – Спектрограма

Система генерує ці спектрограми, тренуючись на навчальних даних, при цьому часто використовують ту ж sequence to sequence архітектуру. Нейронна мережа обробляє записи людського мовлення. Таким чином, вона отримує уявлення про те, як виглядають спектрограми для даного носія мовлення. Seq2seq нейронна мережа узгоджує фонетичні транскрипції з представленнями спектрограми, виведеними з вихідних навчальних даних.

Спектрограма містить числові значення для кожного кадру або тимчасового знімка представленого звуку — і механізму TTS потрібні ці числа для створення голосового аудіофайлу. По суті, модель послідовності відображає текст на спектрограми, які перетворюють текст у числа. Ці числа представляють точні акустичні характеристики голосу того, хто був у записах навчальних даних, якщо цей оратор повинен був вимовити слова, представлені у фонетичній транскрипції. Різні архітектури нейронних мереж використовуються у акустичній моделі для генерації спектрограми.

Найбільш розповсюдженою формою представлення аудіо у задачі синтезу мовлення є мел-спектрограма. Для того, щоб отримати спектрограму, яка відображатиме те, як саме людинаа сприймає звучання, необхідно зробити деякі перетворення, щоб отримати мелчастотні кепстральні коефіцієнти.

Для того, щоб навчити акустичну модель, потрібно пройти кілька етапів. Акустичні моделі дуже вимогливі до якості датасету, тому по-перше, потрібно розмітити дані. Тут потрібно перевірити відповідність тексту аудіо, обрізати межі аудіо(тиша на початку і в кінці запису), проставити наголоси – можна перекласти цю задачу на нейронну мережу, однак з проставленими наголосами результат буде більш якісним, та розкрити абревіатури з транскрипцією та наголосом. Наступний етап – передобробка даних, тут потрібно згенерувати спектрограму, закодувати послідовність вхідного тексту та вилучити додаткові ознаки(морфологія, синтаксис), що може допомогти

нейронній мережі полегшити розстановку інтонацій. Після цього можна проводити навчання моделі.

4.2.1. АРХІТЕКТУРИ АКУСТИЧНИХ МОДЕЛЕЙ

4.2.1.1. Tacotron

У 2017 році Google представив архітектуру нейромережі Tacotron, а через півроку — Tacotron 2. Це модель параметричного синтезу на основі seq2seq архітектури. Вперше вдалося досягти якості, порівнянної з природною людською мовою. Ця модель є лідером серед TTS-систем і займає дуже велику частину ринку. На вхід у модель подається текст для озвучки, при чому вищу якість матиме синтез, якщо попередньо текст буде розбитий на фонеми.

Модель містить кодувальник, механізм Local Sensitive Attention та декодувальник(рисунок 6). Location Sensitive Attention — це механізм уваги, який розширює механізм додаткової уваги, щоб використовувати сукупні вагові коефіцієнти уваги з попередніх часових кроків декодувальника як додаткову функцію.

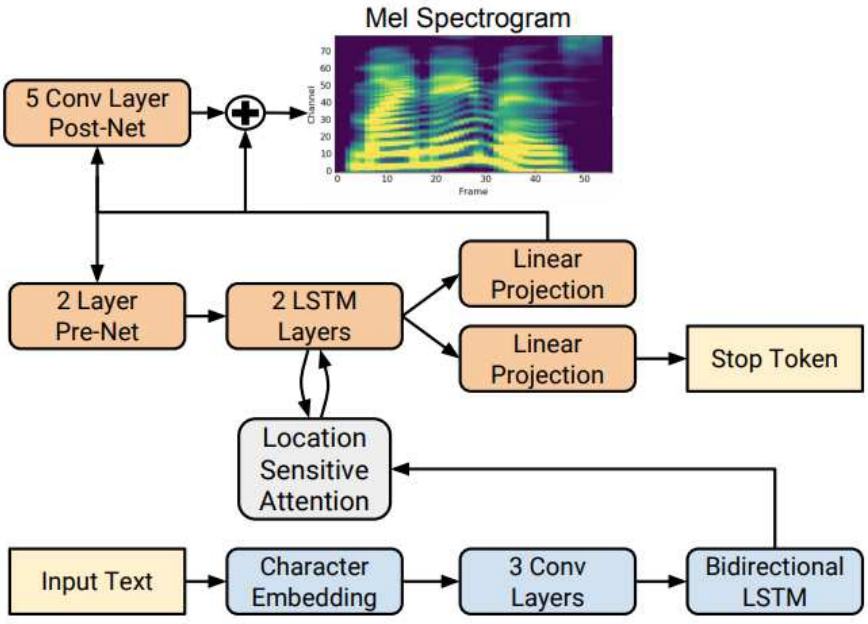

Рисунок 6 – Схема моделі Tacotron

Спочатку ембеддінги літер проходять через кодувальник, що складається з декількох згорткових шарів і двонаправленої LSTM. Додаткові згорткові шари тут потрібні для правильного розрахунку контексту речень. Результати прямого та зворотного проходу LSTM конкатенуються.

Декодер є рекурентною нейронною мережею, тобто кожен наступний крок використовує вихідні дані з попереднього кроку. Тут такі дані це один кадр спектрограми. Ключовим елементом системи є механізм уваги, описаний вище.

Ще одним елементом мережі є PostNet, призначений для покращення спектрограми, створеної декодером.

На виході з декодера отримують спектрограму звукової хвилі, яка передається далі для генерації безпосередньо звукової хвилі.

Модель Tacotron була покращена в наступній модифікації Tacotron 2, яка переробила вихідну архітектуру Tacotron та об'єднала її з вокодером на основі WaveNet.

Перевагами моделі є висока якість, можливість легко додавати нові ознаки. Недоліки архітектури: через рекурентний декодувальник працює повільно, у деяких винятках допускає помилки – як неправильні паузи, так і інтонаційні помилки. Особливо це помітно в питальних реченнях. Також механізм уваги може неправильно працювати на довгих фразах, якщо навчальний датасет містить короткі.

4.2.1.2. FastSpeech

FastSpeech моделі побудовані на архітектурі трансформер. Архітектура трансформер дотримується тієї ж структури кодувальник-декодувальник, але не покладається на рекурентність та згортки для генерування результату. Архітектура моделі трансформер заснована на механізмі уваги, який використовується в архітектурі кодер-декодер у RNN для seq2seq, однак

усуває фактор послідовності. Це означає, що, на відміну від RNN, трансформер обробляє дані не по порядку один за одним, що дає більше можливостей для паралелізму і скорочення часу навчання.

Структуру FastSpeech називають Feed-Forward Transformer(рисунок 7). Feed-Forward Transformer об'єднує кілька блоків FFT для перетворення фонеми в спектрограму з N блоків на стороні фонеми і N блоків на стороні мел-спектрограми, з регулятором довжини між ними. Кожен FFT блок складається з self-attention та одновимірної згорткової мережі, як показано на рисунку 7(b).

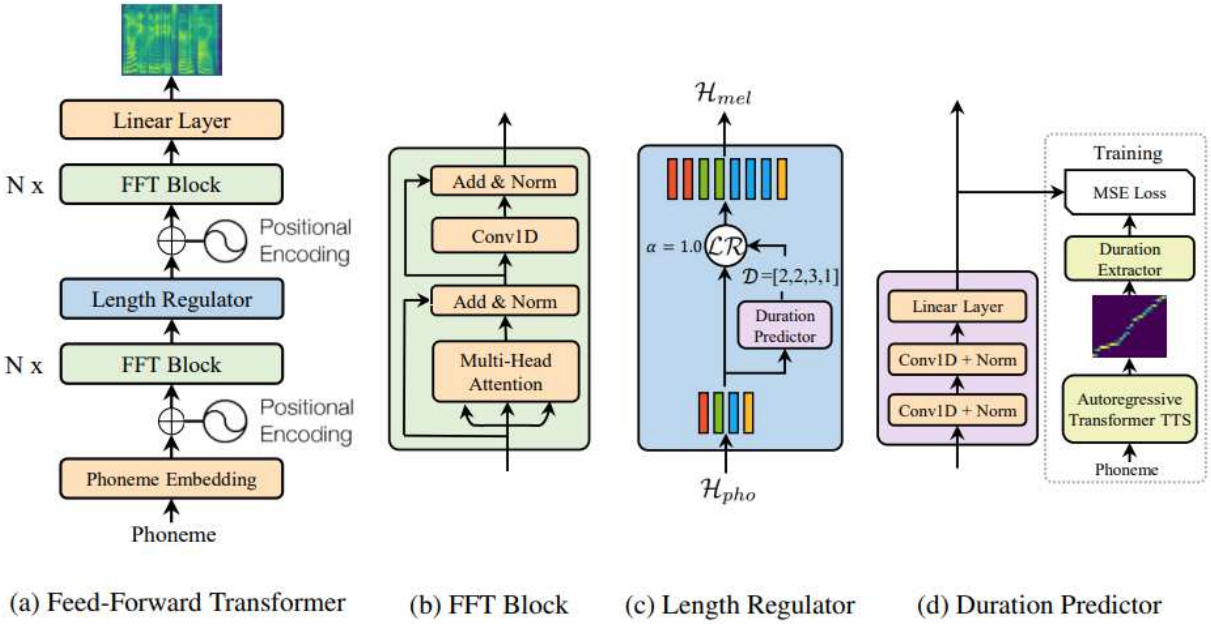

Рисунок 7 – Архітектура FastSpeech

Регулятор тривалості (рис. 7(c)) використовується для вирішення проблеми невідповідності тривалості між фонемою та послідовністю спектрограми в Feed-Forward Transformer, а також для управління швидкістю голосу та частково просодією. Тривалість послідовності фонем зазвичай менша за довжину її послідовності мел-спектрограми, і кожна фонема відповідає кільком мел-спектрограмам.

Для того щоб навчити модель, потрібно для кожного токена отримати тривалість в аудіокадрах для графем або фонем. Це є головним мінусом FastSpeech – потрібна батьківська модель для отримання тривалостей токенів. До того ж, модель обмежена якістю батьківської моделі, оскільки залежна від неї. Ще одним мінусом даної архітектури є те, що для якісного звука потрібно досить багато даних для навчання.

Безумовно плюсами є швидкість навчання та швидкість роботи. Завдяки паралельній генерації мел-спектрограм FastSpeech значно прискорює процес синтезу.

4.2.1.3. Efficient TTS

Efficient TTS оптимізує всі свої параметри за допомогою стабільної, наскрізної процедури навчання, що дозволяє синтезувати високоякісне мовлення швидким та ефективним способом. Efficient TTS мотивується новим підходом моделювання монотонного вирівнювання, який визначає монотонні обмеження для вирівнювання послідовності майже без збільшення обчислень.

Введемо поняття index mapping вектор(IMV). Нехай $\alpha \in R^{(T_1, T_2)}$ – матриця вирівнювання між вхідною послідовністю $x \in R^{T_1}$ та вихідною $y \in R^{T_2}$

Визначимо index mapping вектор(IMV) π як суму індекс вектора $p = [0, 1, \ldots, T_1 - 1]$, зваженим α: $\pi = \sum_{i=0}^{T_1-1} \alpha_{i,j} * p_i$, де $0 \leq j \leq T_2 - 1$, $\pi \in R^{T_2}$, $\sum_{i=0}^{T_1-1} \alpha_{i,j} = 1$. Ми можемо розуміти IMV як очікуване місце для кожного вихідного часового кроку, де очікування є для всіх можливих місць входу в

діапазоні від 0 до $T_1 - 1$. За допомогою IMV реконструюють вирівнювання, використовуючи той факт, що мовлення йде монотонно в одному напрямку.

Загальна схема архітектури зображена на рисунку 8. На етапі навчання(training) ми обчислюємо IMV на основі прихованих представлень текстової послідовності та мел-спектрограми за допомогою генератора IMV. Приховані представлення текстової послідовності та мел спектрограми вивчаються з текстового кодера та мелкодера відповідно. Потім IMV перетворюється на 2-вимірну матрицю вирівнювання, яка використовується для генерування вирівняного за часом представлення через шар реконструкції вирівнювання(Alignment Reconstruction). Подання, вирівняне за часом, передається через декодер, який створює вихідні мел спектрограми. Одночасно ми навчаємо предиктор вирівняної позиції, який навчається передбачати вирівняне положення для кожного маркера введеного тексту. У фазі висновку, ми відновлюємо матрицю вирівнювання з передбачених вирівняних позицій.

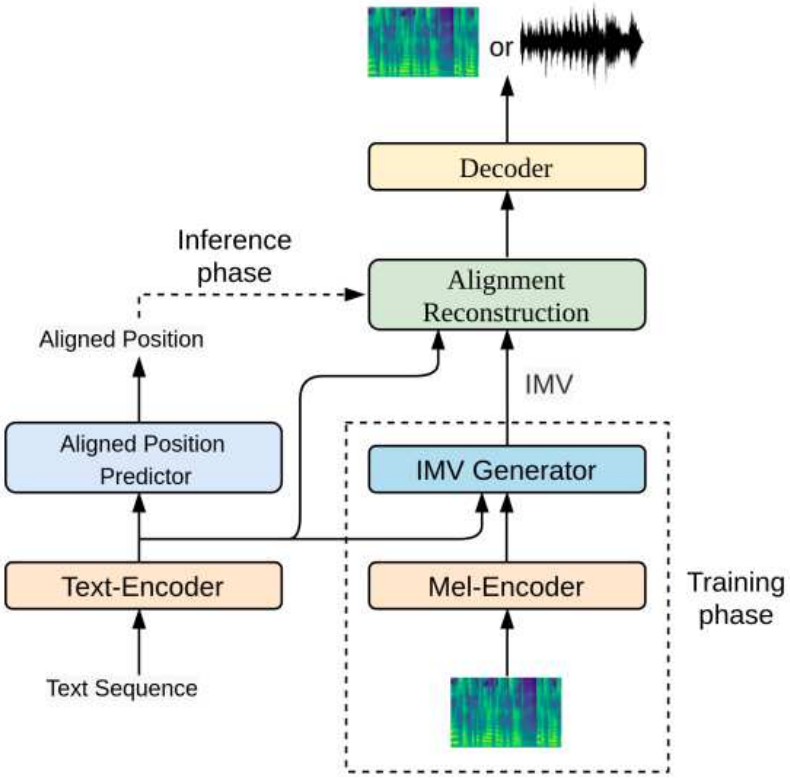

Рисунок 8 – Архітектура моделі Efficient TTS

Перевагами моделі є швидке навчання та дуже швидка робота, вбудований механізм вирівнювання, з мінусів – це все ще нова архітектура.

4.3. СИНТЕЗ ЗВУКУ З СПЕКТРОГРАМИ(ВОКОДЕР)

Тепер залишається перетворити спектрограми в звук. Для цього використовується остання мережа - вокодер. Якщо спектрограми отримують зі звуку за допомогою перетворення Фур'є, то знову отримати звук за допомогою зворотного перетворення не вийде. Гармоніки, які утворюють вихідний сигнал, містять як амплітуду, так і фазу, а наші спектрограми містять лише інформацію про амплітуду. Коли ми робимо зворотне перетворення Фур'є, ми отримуємо поганий звук.

Можна виділити такі типи вокодерів: алгоритмічні, авторегресійні та засновані на GAN.

До алгоритмічних вокодерів належить швидкий алгоритм Гріффіна-Ліма. Він виконує зворотне перетворення Фур'є спектрограми, отримуючи поганий звук, а потім виконує пряме перетворення цього звуку і отримує спектр, який вже містить невелику інформацію про фазу, а амплітуда при цьому не змінюється. Потім знову виконується зворотне перетворення і виходить більш чистий звук. На жаль, якість голосу, виробленого цим алгоритмом, все одно не є високою і звучить неприродною. Перевагами алгоритму є простота реалізації, недоліком, очевидно, невисока якість.

На зміну алгоритму Гріффіна-Ліма прийшли вокодери на основі нейронних мереж, такі як LPCNet, WaveNet, WaveRNN, WaveGlow.

Розглянемо LPCNet вокодер. Його новизна у тому, що він не намагається передбачити складний мовний сигнал безпосередньо з допомогою глибокої нейронної мережі. Натомість нейронна мережа лише

прогнозує менш складний залишковий сигнал голосового тракту, а потім використовує фільтри LPC (Linear Predictive Coding) для перетворення його на остаточний мовний сигнал. На рисунку 9 зображено схему моделі LPCNet. Ліва частина мережі (жовта) обчислюється один раз на кадр, і її результат залишається незмінним протягом усього кадру для мережі частоти дискретизації праворуч. Блок прогнозування обчислення прогнозує вибірку в момент часу t на основі попередніх вибірок і коефіцієнтів лінійного прогнозування.

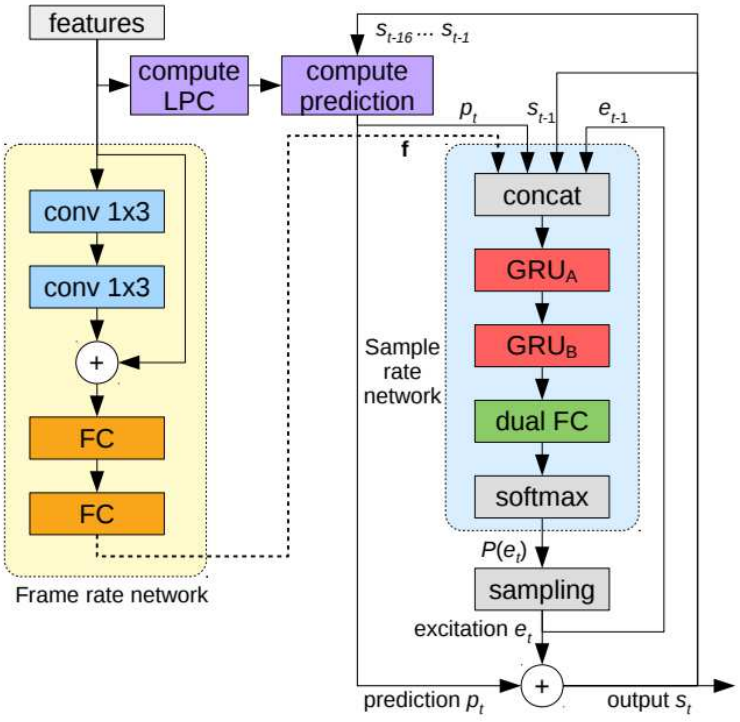

Рисунок 9 – Архітектура моделі LPCNet

LPCNet це досить простий вокодер, який швидко працює та має високу якість. З потенційних недоліків – вокодер потребує тонкого налаштування під деякі голоси.

WaveNet має складнішу архітектуру, і також використовує параметричний підхід. Задача авторегресійного WaveNet, архітектура якогось

заснована на CNN – відновити розподіл ймовірностей звукового сигналу x за допомогою добутку умовних ймовірностей $p(x) = \prod_{t=1}^{T} p(x|\, x_1, x_2 \ldots x_{t-1})$

Модель WaveNet можна розглядати як набір взаємопов’язаних шарів, де кожен вузол у межах шару є комбінацією двох вузлів з попереднього шару.

Основною ідеєю тут є використання причинних згорткових мереж (causal convolution layers) та розширених причинних згорткових мереж (dilated causal convolution layers). Причинна згорткова мережа представляє згорткову нейронну мережу з кількох рівнів, пов'язаних між собою в порядку, який не порушує послідовність вхідного сигналу. Такі мережі навчаються швидше, ніж рекурентні нейронні мережі, але вимагають досить великої кількості рівнів для забезпечення великого signal reception вікна кількості попередніх сигналів, яких залежить оцінка сигналу у даний момент.

Розширені згорткові мережі, які є модифікацією причинних згорткових мереж, здатні збільшити signal reception вікно в рази, що є основною ідеєю моделі WaveNet. Модифікація полягає у застосуванні згортки до області розмірності більшої, ніж її довжина, пропускаючи вхідні зв'язки з деяким кроком. На рисунку 10 зображена схема архітектури WaveNet.

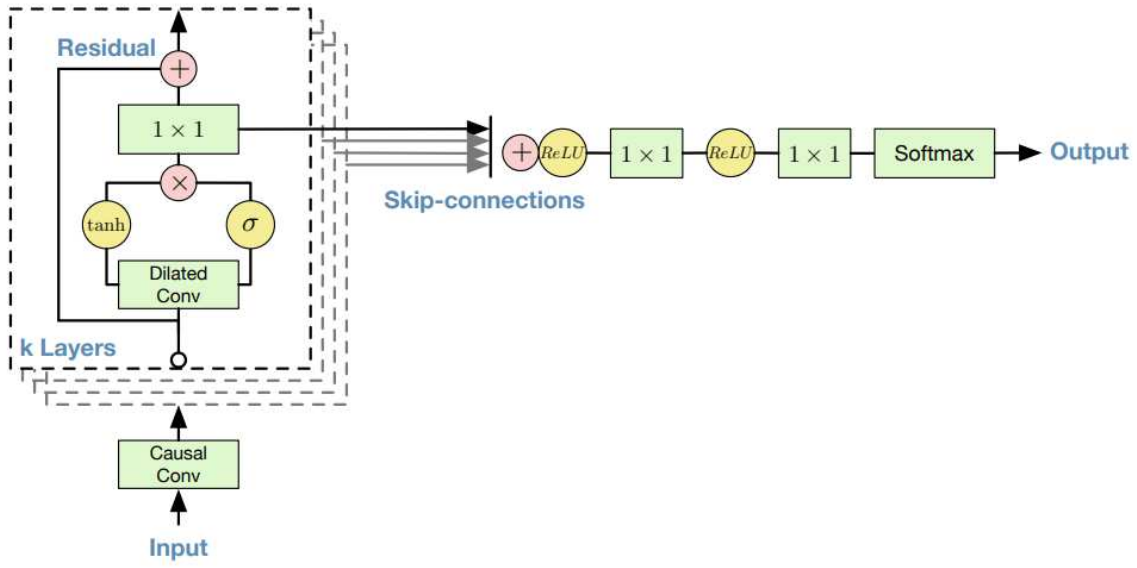

Рисунок 10 – Архітектура WaveNet

До недоліків WaveNet можна віднести складність архітектури та повільну роботу. Синтез відбувається досить довго через авторегресійну природу WaveNet, наприклад, класичний WaveNet генерує 10-секундний сигнал 47 хвилин. Безумовна перевага моделі – висока кінцева якість синтезованого звуку.

Інша реалізація вокодера WaveRNN — це одношарова рекурентна нейронна мережа для генерації аудіо, яка розроблена для ефективного прогнозування 16-бітових необроблених аудіосемплів.
Загальне обчислення в WaveRNN виглядає наступним чином:

$$\mathbf{x}_t = [\mathbf{c}_{t-1}, \mathbf{f}_{t-1}, \mathbf{c}_t]$$

$$\mathbf{u}_t = \sigma(\mathbf{R}_u \mathbf{h}_{t-1} + \mathbf{I}_u^* \mathbf{x}_t)$$

$$\mathbf{r}_t = \sigma(\mathbf{R}_r \mathbf{h}_{t-1} + \mathbf{I}_r^* \mathbf{x}_t)$$

$$\mathbf{e}_t = \tau(\mathbf{r}_t \odot (\mathbf{R}_e \mathbf{h}_{t-1}) + \mathbf{I}_e^* \mathbf{x}_t)$$

$$\mathbf{h}_t = \mathbf{u}_t \cdot \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \cdot \mathbf{e}_t$$

$$\mathbf{y}_c, \mathbf{y}_f = \text{split}(\mathbf{h}_t)$$

$$P(\mathbf{c}_t) = \text{softmax}(\mathbf{O}_2 \text{relu}(\mathbf{O}_1 \mathbf{y}_c))$$

$$P(\mathbf{f}_t) = \text{softmax}\big(\mathbf{O}_4 \text{relu}\big(\mathbf{O}_3 \mathbf{y}_f\big)\big)$$

Кожна частина подається в softmax шар над відповідними 8 бітами. Отриманий шар Dual Softmax дозволяє ефективно прогнозувати 16-бітові вибірки, використовуючи два малих вихідних простори замість одного великого вихідного простору.

Інший підхід – вокодери на основі GAN(generative adversarial network). Генеративна змагальна мережа (GAN) складається з двох частин: генератор та дискримінатор. Генератор вчиться генерувати правдоподібні дані. Згенеровані екземпляри стають негативними навчальними прикладами для дискримінатора. Дискримінатор вчиться відрізняти підроблені дані

генератора від реальних даних та оцінює правдоподібність результатів генератора. Коли починається навчання, генератор видає явно хибні дані, і дискримінатор швидко вчиться це розпізнавати. По мірі навчання генератор наближається до отримання результату, який може переконати дискримінатора. Вокодери на основі GAN мають значний недолік – вони дуже довго навчаються.

Підготовка до навчання вокодера відбувається простіше ніж для акустичної моделі – тут потрібно тільки згенерувати спектрограми, після чого навчати модель. При навчанні моделей часто виникають проблеми. Найголовніша з них – відсутність метрик для зупинки навчання, середнє відхилення точно не показує якість звуку, тому часто доводиться слухати згенеровані результати, і на основі цього приймати рішення про зупинку. Інша проблема, спричинена відсутністю метрик – не працює валідація на відкладеній виборці, тут також часто потрібно прослухати досить велику кількість згенерованих даних, щоб мати розуміння про якість.

5. РОЗВИТОК СИНТЕЗУ МОВИ

Нейронний синтез мови – відносно молода галузь. Зараз сфера синтезу мови активно розвивається(рисунок 11), кожен рік з'являються нові реалізації, які оптимізують попередні, або адаптують їх до своїх потреб.

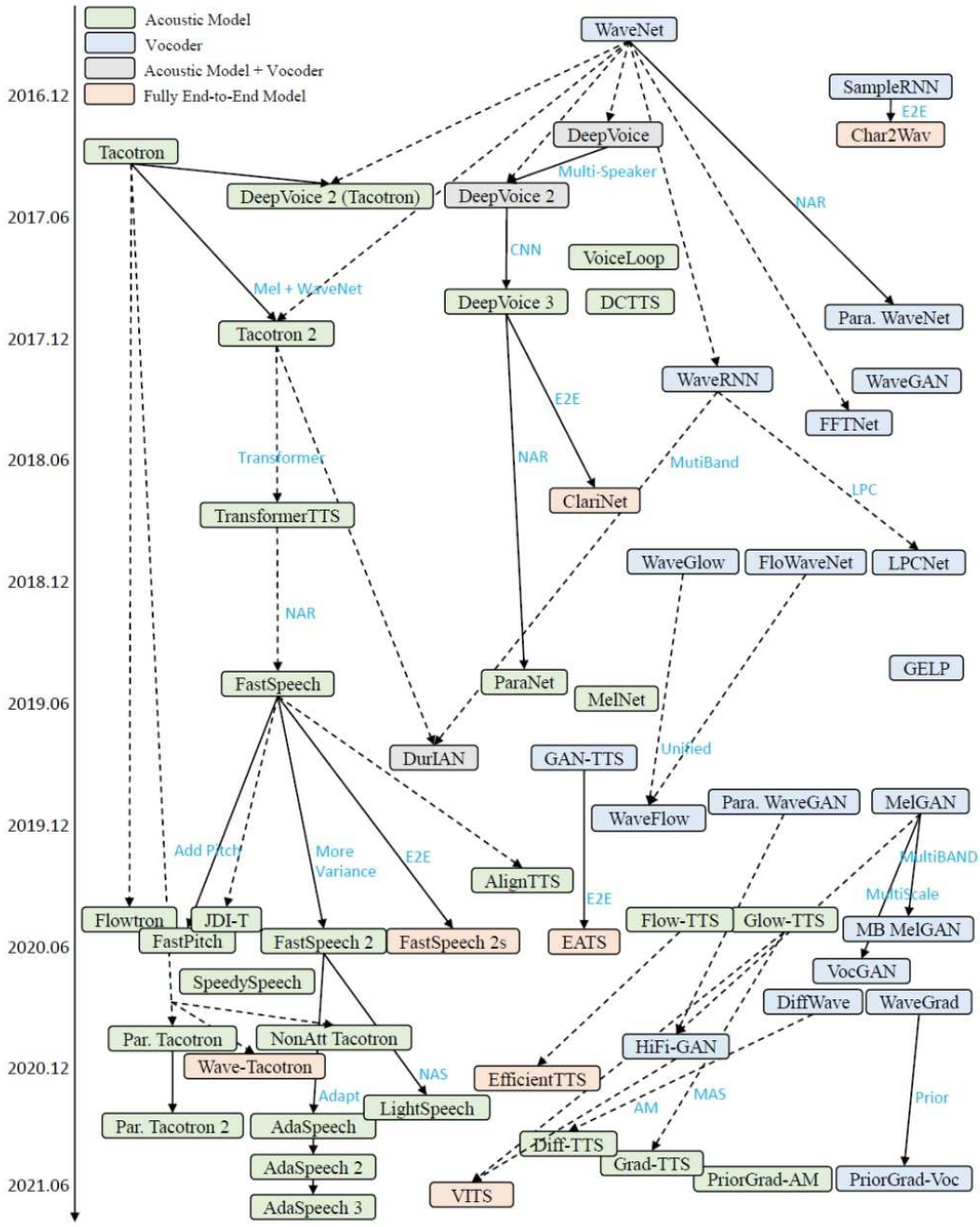

Рисунок 11 – Розвиток моделей синтезу мови

Реалізаціям систем синтезу мови є куди розвиватись – існують ще не до кінця вирішені задачі. Зараз синтез мови розвивається по декільком

напрямкам. Один з напрямків – реалізація мультиспікера та синтез неіснуючих голосів – тут використовуються, наприклад, варіаційні автокодувальники. Системи з кількома носіями мови потребують багато навчальних даних, і поки такі системи синтезу високої якості не розвинені. Приклад системи, що генерує неіснуючий голос це TacoSpawn – рекурентна модель з увагою, яка розширює модель Tacotron.

Цікавою задачею також є багатомовний синтез, а саме випадок генерації одним голосом мовлення на різних мовах.

Ще один напрямок – вираження емоцій в синтезованому мовленні, причому не тільки дискретних, а і плавних переходів між емоційними станами. Тут також використовують варіаційні автокодувальники, однак ця задача поки залишається відкритою.

Схожа проблема – монотонність і відсутність експресії та інтонацій у синтезованій мові та регулювання тону. Не можна передбачити, яка фраза буде вимовлена весело та оптимістично, а яка – грубо чи самовпевнено. Іноді компенсується можливістю налаштування інших мовних тонкощів, однак все ще постає задача генерації мовлення з різними інтонаціями, які можна регулювати і контролювати.

Ще одна проблема у даній сфері - визначення якості синтезованої мови. Визначення якості не має автоматизованого або тривіального рішення. Якість вимірюється на основі багатьох факторів таких як природність, надійність, правильність, зрозумілість. Оскільки для визначення продуктивності моделі необхідно перевірити якість, це повинні робити люди. Людей-асесорів, які розмовляють мовою, яку потрібно оцінити, просять оцінити якість звуку звукового сигналу. Середня оцінка думки, mean opinion score (MOS) розраховується на основі балів, отриманих шляхом опитування оцінки якості (не стандартизованої, а тому суб’єктивної) відтворення звуку. Оцінка варіюється від 1 для поганого до 5 для відмінного звуку. Вибраним слухачам

пропонується прослухати згенеровані аудіофайли, іноді в порівнянні з вихідними файлами. Після прослуховування вони ставлять оцінку, а середнє значення балів дає оцінку MOS. На рисунку 12 оцінки MOS для відомих систем синтезу мови.

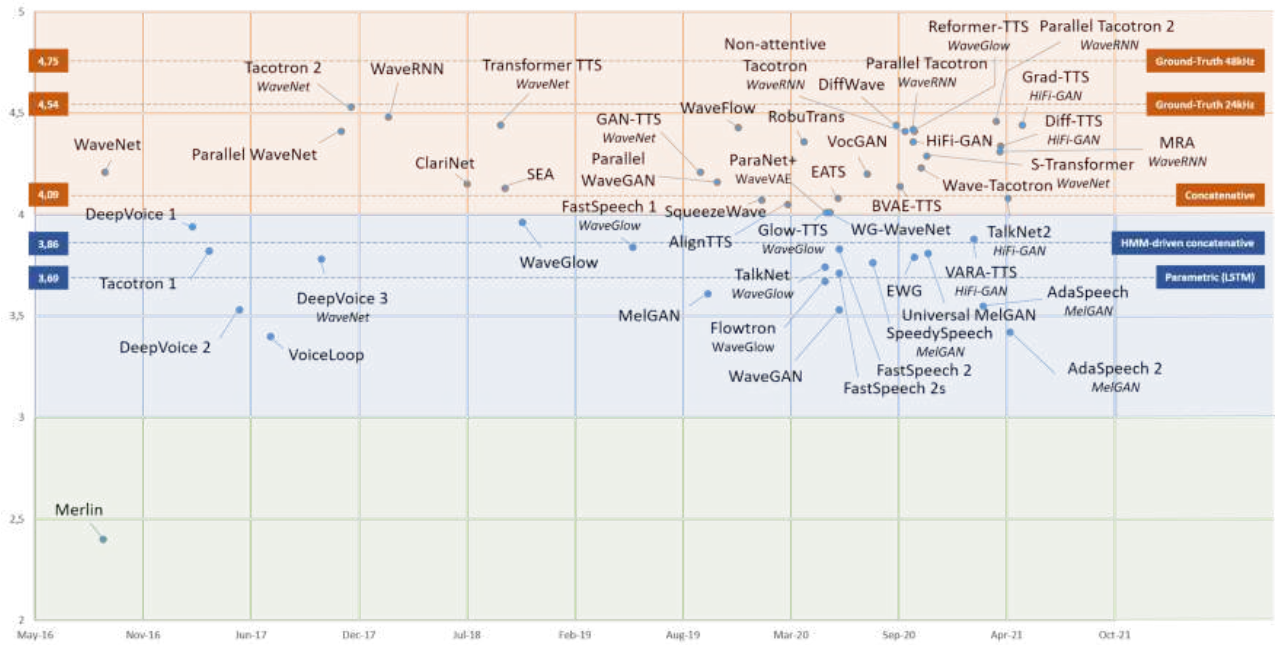

Рисунок 12 – MOS оцінки деяких TTS систем

Звідси, наприклад, можна побачити, що реалізація Tacotron 2 з WaveNet вокодером показує значно вищу якість, ніж Tacotron 1 на основі алгоритму Гріффіна-Ліма. Оцінку ж розглянутої раніше архітектури FastSpeech можна побачити на рисунку 13.

Method	MOS
GT	4.30 ± 0.07
GT (Mel + PWG)	3.92 ± 0.08
Tacotron 2 (Shen et al., 2018) (Mel + PWG)	3.70 ± 0.08
Transformer TTS (Li et al., 2019) (Mel + PWG)	3.72 ± 0.07
FastSpeech (Ren et al., 2019) (Mel + PWG)	3.68 ± 0.09
FastSpeech 2 (Mel + PWG)	3.83 ± 0.08
FastSpeech 2s	3.71 ± 0.09

Рисунок 13 – MOS оцінка Tacotron, FastSpeech

Як бачимо, FastSpeech модель у якості не дуже поступається лідеру ринку Tacotron, та є значно швидшою.

ВИСНОВОК

В рамках даної роботи були досягнуті наступні результати:

- Досліджені основні підходи до вирішення задач нормалізації, обробки текстів, синтезу спектрограм та розглянуті вокодери.
- Були проаналізовані переваги та недоліки відомих архітектур акустичних моделей та вокодерів.
- Наведено сучасний стан розробки систем синтезу мовлення, відкриті задачі та проблеми.

Системи синтезу мови пройшли шлях від конкатенативного підходу та умовного алгоритму Гріффіна-Ліма до використання великої кількості нейронних мереж на всіх етапах синтезу.

В процесі порівняння та аналізу систем було визначено, що кожен підхід має свої позитивні та негативні сторони, і вибір архітектури багато в чому залежить від наявних ресурсів, часу, об’єму набору даних, мови синтезу, сфери застосування, але без сумнівів можна сказати, що системи синтезу мови постійно проходять оптимізацію по різним параметрам, покращуються та адаптуються до нових задач.

СПИСОК ЛІТЕРАТУРИ

1. https://www.upgrad.com/blog/types-of-neural-networks/ [Електронний ресурс]
2. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks [Електронний ресурс]
3. https://www.respeecher.com/blog/what-is-text-to-speech-tts-initial-speech-synthesis-explained [Електронний ресурс]
4. https://wiki.aalto.fi/display/ITSP/Concatenative+speech+synthesis [Електронний ресурс]
5. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.742.2786&rep=rep1&type=pdf [Електронний ресурс]
6. https://www.cs.cmu.edu/~awb/papers/IEEE2002/allthetime/node1.html [Електронний ресурс]
7. https://kwantics.com/how-speech-synthesis-works-complete-guide/ [Електронний ресурс]
8. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Електронний ресурс]
9. ХОМИЦЕВИЧ, Ольга Гурьевна; РЫБИН, Сергей Витальевич; АНИЧКИН, Илья Михайлович. Использование лингвистического анализа для нормализации текста и снятия омонимии в системе синтеза русской речи. Известия высших учебных заведений. Приборостроение, 2013, 56.2: 42-46.
10. https://bastings.github.io/annotated_encoder_decoder/ [Електронний ресурс]
11. YOLCHUYEVA, Sevinj; NÉMETH, Géza; GYIRES-TÓTH, Bálint. Grapheme-to-phoneme conversion with convolutional neural networks. Applied Sciences, 2019, 9.6: 1143.

12. WANG, Yuxuan, et al. Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135, 2017.
13. REN, Yi, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558, 2020.
14. MIAO, Chenfeng, et al. Efficienttts: An efficient and high-quality text-to-speech architecture. In: International Conference on Machine Learning. PMLR, 2021. p. 7700-7709.
15. OORD, Aaron van den, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.
16. https://habr.com/ru/company/sberdevices/blog/548812/ [Електронний ресурс]
17. VALIN, Jean-Marc; SKOGLUND, Jan. LPCNet: Improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019. p. 5891-5895.
18. OKAMOTO, Takuma, et al. Real-Time Neural Text-to-Speech with Sequence-to-Sequence Acoustic Model and WaveGlow or Single Gaussian WaveRNN Vocoders. In: INTERSPEECH. 2019. p. 1308-1312.
19. https://towardsdatascience.com/text-to-speech-foundational-knowledge-part-2-4db2a3657335 [Електронний ресурс]