

DOI: <https://doi.org/10.36910/6775-2524-0560-2023-50-03>

УДК 004.93

Сіромаха Аліна Геннадіївна, магістр

П'ятикоп Олена Євгенівна, к.т.н., доцент

<https://orcid.org/0000-0002-7731-3051>

Котихова Людмила Дмитрівна, асистент

<https://orcid.org/0009-0006-5008-622X>

ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, Україна

## ПИТАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ПРОВЕДЕННЯ ТА ПЕРЕВІРКИ ДИКТАНТІВ

Сіромаха А.Г., П'ятикоп О.Є., Котихова Л.Д. Питання розробки програмного забезпечення для автоматизації проведення та перевірки диктантів. Раніше диктанти були невід'ємною частиною будь-якого навчання та вивчення мови, для кращого запам'ятовування слів та орфографії. Але на цьому етапі життя через різні зовнішні причини навчання часто вимушено бути дистанційним. В цих умовах стало складніше проводити даний вид роботи, тому написання диктантів майже припинило своє існування, хоча це дуже ефективна форма роботи. Розвиток інформаційних технологій дозволяє створити програмне забезпечення для автоматизації процесу проведення диктантів. В статті наведено аналіз публікацій щодо синтезу мови та пошуку подібності між текстами. Надано опис прототипу програми, яка проводить вимову тексту та подальшу перевірку набраного користувачем тексту з початковим. В статті приводяться результати тестування програми для різних видів помилок.

**Ключові слова:** диктант, орфографія, синтезатор мови, автоматизоване порівняння тексту, перевірка синтаксису.

**Siromakha A.G., Piatykop O.Y., Kotykhova L.D. Issues of developing software for automating the conduct and verification of dictations.**

Previously, dictations were an integral part of any language learning and learning for better memorization of words and spelling. But at this stage of life, due to various external reasons, education is often forced to be remote. Under these conditions, it became more difficult to carry out this type of work, so writing dictations almost ceased to exist, although this is a very effective form of work. The development of information technology makes it possible to create software for automating the process of conducting dictations. The article presents an analysis of publications in the field of language synthesis methods and the search for similarities between texts. A description of the prototype of the program is presented, which performs the pronunciation of the text and the subsequent verification of the text typed by the user with the original one. The article presents the results of testing the program for different types of errors.

**Keywords:** dictation, spelling, language synthesizer, text comparison automation, syntax check.

**Постановка наукової проблеми.** Раніше диктанти були невід'ємною частиною будь-якого навчання та вивчення мови, для кращого запам'ятовування слів та орфографії. Але на цьому етапі життя через зовнішні причини і зручність перебування на дому або будь-де люди почали переходити на дистанційне, тобто онлайн навчання, де стало складніше проводити даний вид роботи, тому написання диктантів майже припинило своє існування, хоча це дуже ефективна форма роботи.

Нині сучасні інформаційні технології розвиваються дуже швидко. Створюється безліч різних програмних засобів для поліпшення і спрощення життя людей.

З початку 50-х років минулого століття інтенсивні дослідження сприяли створенню першої системи формантного синтезу мови. У наступні два десятиліття дослідження в цій галузі породили безліч робіт з базових методів лінгвістичної та просодичної обробки. Проте сам синтез мови не виходив за межі формантного підходу [1].

З поширенням комп'ютерів, початок 90-х було ознаменовано новим підходом до генерування промови – Unit–Selection (компіляційний синтез). Синтез мови за допомогою Unit–Selection кардинально відрізнявся від формантного синтезу високою якістю та природністю. Для максимального наближення синтезованої мови до природної людської за допомогою Unit Selection потрібно мати великий мовний корпус та серйозні обчислювальні потужності. Однак цей підхід мав свої недоліки, крім високої вартості, наприклад низьку гнучкість. Фактично було неможливо було змінювати акустичні ознаки під час синтезу мови для покращення експресивності мови або зміни її емоційності [2].

Кінець 90-х і початок 2000-х став свідком нового підходу до генерування промови з тексту, відомим як статистичний параметричний синтез промови (скорочено СПСМ). СПСМ вирішує основні проблеми, з якими зіткнувся Unit–Selection. Ця технологія дозволяє змінювати акустичні ознаки під час синтезу мови, вимагає невеликого обсягу місця для зберігання даних, і дуже зручна для дослідження впливу акустичних параметрів на емоційність мови.

До початку 2010-х технологія прихованих марківських моделей (скорочено СММ) домінувала у системах СПСМ завдяки низці переваг, зокрема, таких як надійність та швидкість обчислення. Однак, незважаючи на те, що технологія СММ працює досить ефективно, вона має серйозні обмеження в точності передбачення акустичних параметрів, оскільки, як показала практика, дерева прийняття рішень неефективні для моделювання складних контекстних залежностей.

Застосування глибокої нейронної мережі (англійською – Deep Neural Network, скорочено DNN) в СПСМ показало, що нейронні мережі можуть досягти більшої природності мови, ніж традиційні СММ, навіть з однаковою кількістю параметрів [3].

Пізніше у 2015 році Хейга Зен першим застосував нейронні мережі з довгою короткостроковою пам'яттю (англійською – Long-Short Term Memory, скорочено LSTM), завдяки чому акустичне моделювання в СПСМ перейшло від покадрового на ефективніше моделювання за послідовністю.

Однак мовні параметри, що генеруються цими моделями, виходять сильно згладженими, внаслідок чого якість мови, як і раніше, залишається невисокою в порівнянні з природною мовою. Ефект сильно згладжених акустичних параметрів є однією із найскладніших проблем у синтезі мови.

Також стало очевидним, що ступінь наукового опрацювання таких технологій дуже відрізняється для різних мов. Одним із основних та важливих напрямів є розробка високоякісних методів просодичної та лінгвістичної обробки для малоресурсних мов, як зазначено у статті [4].

**Метою роботи** є створення програмного забезпечення для автоматизації проведення та перевірки диктантів з використанням синтезу мови та методів порівняння текстових документів для пошуку помилок (виявлення текстуальної подібності).

#### **Аналіз останніх досліджень і публікацій.**

Все частіше у повсякденному житті стали використовувати синтезатори мови. Як видно вже за однією назвою, вони здійснюють синтез мови, тобто формують письмовий текст в усний. За науковим визначенням – це комп'ютерне моделювання людської мови з текстового подання за допомогою методів машинного навчання. Завдяки цьому, наприклад, можна вивчати нові іноземні слова з правильною вимовою, читати книги не відволікаючись від своїх справ, sms-повідомлення, документи та цілі веб-сайти, можна будувати запити в пошукових системах без допомоги клавіатури, отримувати інформацію з особистого рахунку без використання паролів і навіть давати вказівки персональному автомобілю чи озвучувати текст диктантів на уроках у школах та на екзаменах, що частково виконується в даній роботі.

Існує кілька методів синтезу мови. Перший полягає у побудові слів шляхом комбінування фонем і алофон. Отримана фонемна послідовність, після вибору висоти тону та інтонації, перетворюється на мовлення. При такому підході генерується цілком зрозуміла мова, але користувачу, що слухає, ясно, що їх вимовляє робот. Другий метод синтезу мови полягає у її імітації з використанням моделі голосового тракту людини. Тут глухі і дзвінки згодні в мові є джерелами періодичних і шумових сигналів відповідно. Потім сигнали проходять крізь каскад фільтрів [5].

Існує безліч статей, де розглядаються різні методи використання синтезатору мови та його вдосконалення.

У статті [6] розбирається алгоритм стеммінгу, процедура скорочення всіх слів з однаковим до загальної форми, корисний у багатьох галузях обчислювальної лінгвістики та інформаційно-пошукова робота. Хоча форма алгоритму варіюється в залежності від його застосування, деякі лінгвістичні проблеми є спільними для будь-якої процедури стеммінгу. Як основу для оцінки попередніх спроб із цими проблемами, у цій статті спочатку обговорюються теоретичні та практичні атрибути алгоритмів стеммінгу. Потім нова версія контекстно-залежної запропонований алгоритм найбільшої відповідності для англійської мови; хоча розроблена від використання у системі передачі бібліотечної інформації, вона має загальний характер. Основна лінгвістична проблема у виділенні коріння, варіації у написанні стебел, обговорюється досить докладно.

У статті [5] пропонується нейронне збудження на основі WaveNet. Модель (ExcitNet) для статистичного параметричного синтезу промови системи. Звичайні системи нейронного вокодування на основі WaveNet значно покращують якість сприйняття синтезованої мови шляхом статистичної генерації тимчасової послідовності мовних сигналів через авторегресивну структуру. Поліпшити ефективність моделювання, запропонований вокодер ExcitNet використовує адаптивний зворотний фільтр для відокремлення спектральних складових від мовного сигналу. Залишкова

складова, тобто сигнал збудження. Потім навчені та згенеровані в рамках WaveNet. Таким чином, якість синтезованого мовного сигналу може бути надалі покращена, оскільки спектральна складова добре представлена фреймворком глибокого навчання і, крім того, залишковим компонентом, що ефективно генерується структурою WaveNet. Експериментальні результати показують, що запропонований вокодер ExcitNet, навчається в залежності від того, хто говорить, так і незалежно від того, хто говорить, перевершує традиційні вокодери з лінійним передбаченням і аналогічним чином налаштовані звичайні вокодери WaveNet.

У статті [7] представлена ефективність систем перетворення тексту на мову у поєднанні з нейронними вокодерами для створення високоякісної мови. Проте збирання необхідних навчальних даних та створення цих передових систем з нуля — це час і ресурсомісткий. Економічний підхід полягає у розробці нейронних вокодерів для поліпшення мови, що генерується існуючими та недорогими системами TTS. Проте такий підхід зазвичай достатній. Включає 2 проблеми: 1) тимчасові невідповідності між TTS та природними формами хвиль 2) акустичні невідповідності для перевірки та тестування.

Для вирішення цих проблем приймають циклічну модель перетворення голосу, для створення узгоджених у часі даних, для навчання та акустично узгоджені покращені дані для тестування нейронних вокодерів. Ця структура може бути застосована до довільних систем TTS та нейронних вокодерів. У цій роботі застосовується метод із сучасним вокодером WaveNet для двох різних базових систем TTS, а також об'єктивні та суб'єктивні експериментальні результати, що підтверджують ефективність запропонованої структури.

У статті [8] пропонують реалізувати покращення мови шляхом регенерації чистої мови з «помітного» уявлення, витягнутого з шумного сигналу. Мережа, яка виділяє суттєві ознаки, навчається, використовуючи набір клонів мережі екстрактора з поділом ваги. Клони отримують спектри різних шумових версій, той самий мовний сигнал, як і вхід. За висновком, суттєві особливості формують вхідні дані для мережі WaveNet, які генерують природний і чистий мовний сигнал з тими самими атрибутами, що і чистий наземний сигнал. У міру того, як сигнал стає більш шумним, система робить природні звукові помилки, які залишають на мовленні різноманіття замість традиційних артефактів, знайдених в інших системах.

Як стверджують у цій статті, проведені експерименти підтверджують, що система генеративного покращення забезпечує сучасну продуктивність покращення за подібним тестом. Система на основі клонів відповідає або перевершує інші системи за кожним вхідним діапазоном відношення – сигнал–шум зі статистичною значимістю.

У статті [9] робиться спроба визначити переваги двомовного навчання. Використовується мова trix (казахська) та вбудована мова (російська). Зокрема, оцінюється продуктивність моделі на матричних словах та вкладеннях виділених слів окремо. Використовують два набори даних: казахська мова з перемиканням коду та російська без перемикання коду. Навчають одномовну модель для кожного набору даних та двомовну модель для їхнього поєднання.

Основна мета експериментів — порівняти продуктивність моделі, навченої мови з перемиканням коду, з моделлю, навченої мови без перемикання коду, тобто на висловлюваннях обох текстів. Експериментальні результати показують, що двомовне навчання покращує навички моделі продуктивності на матричних словах та значно покращує її продуктивність на вбудовані слова. Спостерігається покращення 14,69% у кодових словах.

Також, в даний час має велике практичне застосування пошук подібності між текстами, у тому числі для виявлення плагіату в наукових та творчих дослідженнях. Існує три основні категорії виявлення текстуальної подібності: порівняння на основі слів, лінійний пошук на основі пунктів, що використовується пошуковими системами та стилістичний аналіз. Також існують методи, засновані на різних характеристиках текстів, такі як методи, що ґрунтуються на семантиці, як для виявлення плагіату; так і для пошуку інформації.

На вхіді процесу порівняння текстів є два документи, призначені для порівняння, один із яких є еталоном. На першому рівні аналізу проводиться вилучення текстових пасажів, виходом цього першого рівня буде перелік значних пасажів з кожного документа, які послужать як вхідні дані для наступного рівня для дозволу анафори, а останній, своєю чергою, надходить до рівня уявлення схем. Побудована схема подання є входом для виявлення рівня схожості між текстовими пасажами.

На цьому рівні, як вхідні дані використовуються текстові пасажі з документів і на основі їх результатів порівняння, виконується обчислення подібності між цими двома документами.

**Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.** Для реалізації програмного забезпечення для автоматизації проведення та перевірки диктанту «Dictation Practice», було визначено умови, що повинні враховуватись при розробці:

- повинна працювати на базі операційної системи Windows;
- програма повинна мати процес перетворення тексту на мовлення для диктування;
- необхідно мати такі можливості як зупинка відтворення тексту та його включення спочатку;
- можливість вибору різних бібліотек синтезу мови, для вибору зручної для конкретної людини;
- можливість перевірки тексту, порівняння з початковим та виведення їх на екран;
- повинен у кінці виводитись висновок результатів написання диктанту, таких як «кількість слів написаних всього», «кількість знаків пунктуації», «кількість помилок», а також виділення помилок у самому тексті.

За ними було побудовано діаграму використання за методологією UML, що зображено на рисунку 1.



Рис. 1 – Діаграма варіантів використання додатку «Dictation Practice»

При розробці було використано бібліотеки для створення інтерфейсу користувача Win Forms (Windows Forms) та .NET Framework, а сам модуль написано на високорівневій мові програмування C#. Причиною використання є швидкість роботи .NET Framework та мови C#, та можливість формування користувацьких елементів керування у бібліотеці Win Forms. В якості синтезаторів мови використано бібліотеки Microsoft, Google, Amazon.

Розглянемо класи додатку «Dictation Practice» докладніше:

- OpenFileForm – початкова форма додатку, у якій необхідно обрати вже існуючий у додатку файл чи додати новий з комп'ютера, для початку роботи з вчиткою диктанту;
- MainForm – головна форма додатку, де є можливість вибору бібліотеки синтезатору мови (Microsoft, Google, Amazon), а також програвання тексту та форма для запису диктанту учнем для подальшої перевірки та виводу результату;
- DictationForm – кінцева форма, у якій вже виводиться результат перевірки / порівняння двох текстів та пошуку помилок.

Загальну діаграму класів наведено на рисунку 2.

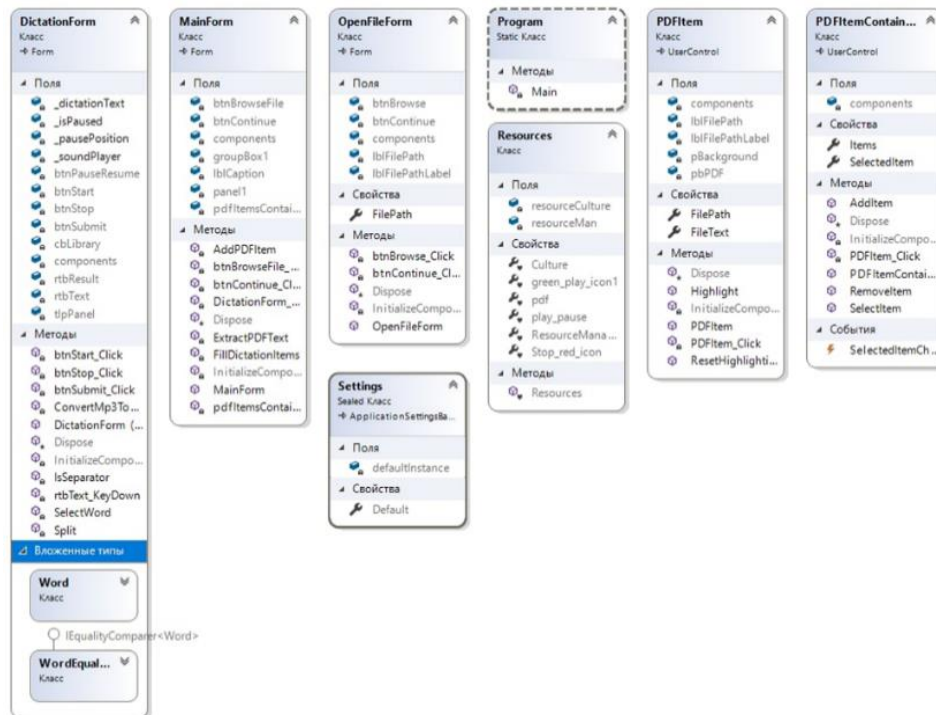


Рис. 2. Діаграма класів програми проведення диктантів «Dictation Practice»  
Перевірка дефектів на практиці у розрахунковому вигляді надана у таблиці 1.

Таблиця 1. Результати використання програми проведення диктантів «Dictation Practice»

№	Можливі дефекти	Вхідні дані	Вихідні дані
1	Орфографічні помилок (по 1 в слові по всьому тексті)	8	8
2	Орфографічні помилки (по декілька в 1 слові по всьому тексті)	14	9
3	Пропуск знаку пунктуації	6	Розподілених знаків: 5 з 11
4	Зайві знаки пунктуації	15	Розподілених знаків: 15 з 11
5	Пропуск потрібного знаку пунктуації та занесення зайвого	Пропущено 6, додано 4	Розподілених знаків: 9 з 11
6	Написання усіх слів малими буквами	Задано усі слова (78)	4 помилки у словах, де потребується велика буква
7	Написання усіх слів великими буквами	Задано усі слова (78)	61 помилка (виділяє майже всі слова)
8	Використання непотрібних пробілів	Додано 9	Не сприймає як помилку (0)
9	Пропуск пробілів (не склеюючи слова)	7	Немає помилок, кількість слів 78 з 78
10	Пропуск пробілів (склеюючи по два слова)	7	Кількість слів 71 з 78 (сприймає як 1 слово), помилок 14 (на кожне слово)
10	Пропуск пробілів (склеюючи по декілька слів)	Пропущено 5 пробілів (склеюючи у 1 слово)	Кількість слів 73 з 78 (сприймає як 1 слово), помилок 5 (на кожне слово)
12	Використання спецсимволів (не склеюючи зі словом)	2	Кількість слів 80 з 78, помилок 0
13	Використання спецсимволів (склеюючи зі словом)	2	Кількість слів 78 з 78, помилок 2

**Висновки та перспективи подальшого дослідження.** Нині сучасні інформаційні технології розвиваються дуже швидко. Створюється безліч різних програмних засобів для поліпшення і спрощення життя людей. Особливо це актуально в умовах дистанційного навчання. В умовах

онлайн навчання традиційна форма проведення диктантів потребує трансформації. Тому можна запропонувати автоматизацію процесу проведення та перевірки диктантів за допомогою відповідного програмного забезпечення. В роботі проведено аналіз наукових досліджень та публікацій сучасного стану технологій синтезу мови та перевірки подібності тексту. За допомогою мови програмування C# та бібліотек синтезу мови Microsoft, Google, Amazon розроблено програмне забезпечення, яке дозволяє користувачу прослухати та набрати текст, а потім автоматично перевірити його з початковим зразком. Проведене тестування на різні види помилок показало можливість та доцільність розробки. Наступна діяльність буде присвячено покращенню ефективності роботи програми. Використання подібних програм сприятиме удосконаленню організації проведення та перевірки диктантів під час дистанційної форми навчання та для самопідготовки учнів.

#### Список бібліографічного опису

1. Fant G. Acoustic theory of speech production, The Hague, The Netherlands, Mouton, 1960.
2. Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. Class-based n-gram models of natural language // Computational linguistics. 1992. V. 18. P.467-479.
3. Stratos K. et al. A Spectral Algorithm for Learning Class-Based n-gram Models of Natural Language //UAI. – 2014. – P. 762-771.
4. Kaliyev A., Rybin S.V., Matveev Y. The pausing method based on brown clustering and word embedding // Lecture Notes in Computer Science. 2017. V. 10458. P. 741-747
5. Morise M., Yokomori F., Ozawa K. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications //IEICE TRANSACTIONS on Information and Systems. – 2016. – Т. 99. – №. 7. – С. 1877-1884.
6. Lovins Julie Beth. Development of a Stemming Algorithm // Mechanical Translation and Computational Linguistics. — 1968.
7. Morise M., Watanabe Y. Sound quality comparison among high-quality vocoders by using re-synthesized speech //Acoustical Science and Technology. – 2018. – Т. 39. – №. 3. – С. 263-265.
8. Watts O., Valentini-Botinhao C., King S. Speech Waveform Reconstruction Using Convolutional Neural Networks with Noise and Periodic Inputs //ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2019. – С. 7045-7049.
9. Dmitrii Ubskii, Yuri Matveev, Wolfgang Minker. Impact of Using a Bilingual Model on Kazakh Code-Switching Speech // «Grammatical Structure in Codeswitching». 2019. P. 6

1. ite DK, Tudor-Locke C, Zhang Y, Fielding R, LaValley M, Felson DT, et al.
2. Daily walking and the risk of incident functional limitation in knee
3. osteoarthritis: an observational study. Arthritis Care Res (Hoboken).
4. 2014;66(9):1328-

#### References

1. Fant G. Acoustic theory of speech production, The Hague, The Netherlands, Mouton, 1960.
2. Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. Class-based n-gram models of natural language // Computational linguistics. 1992. V. 18. P.467-479.
3. Stratos K. et al. A Spectral Algorithm for Learning Class-Based n-gram Models of Natural Language //UAI. – 2014. – P. 762-771.
4. Kaliyev A., Rybin S.V., Matveev Y. The pausing method based on brown clustering and word embedding // Lecture Notes in Computer Science. 2017. V. 10458. P. 741-747
5. Morise M., Yokomori F., Ozawa K. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications //IEICE TRANSACTIONS on Information and Systems. – 2016. – Т. 99. – №. 7. – С. 1877-1884.
6. Lovins Julie Beth. Development of a Stemming Algorithm // Mechanical Translation and Computational Linguistics. — 1968.
7. Morise M., Watanabe Y. Sound quality comparison among high-quality vocoders by using re-synthesized speech //Acoustical Science and Technology. – 2018. – Т. 39. – №. 3. – С. 263-265.
8. Watts O., Valentini-Botinhao C., King S. Speech Waveform Reconstruction Using Convolutional Neural Networks with Noise and Periodic Inputs //ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – IEEE, 2019. – С. 7045-7049.
9. Dmitrii Ubskii, Yuri Matveev, Wolfgang Minker. Impact of Using a Bilingual Model on Kazakh Code-Switching Speech // «Grammatical Structure in Codeswitching». 2019. P. 6